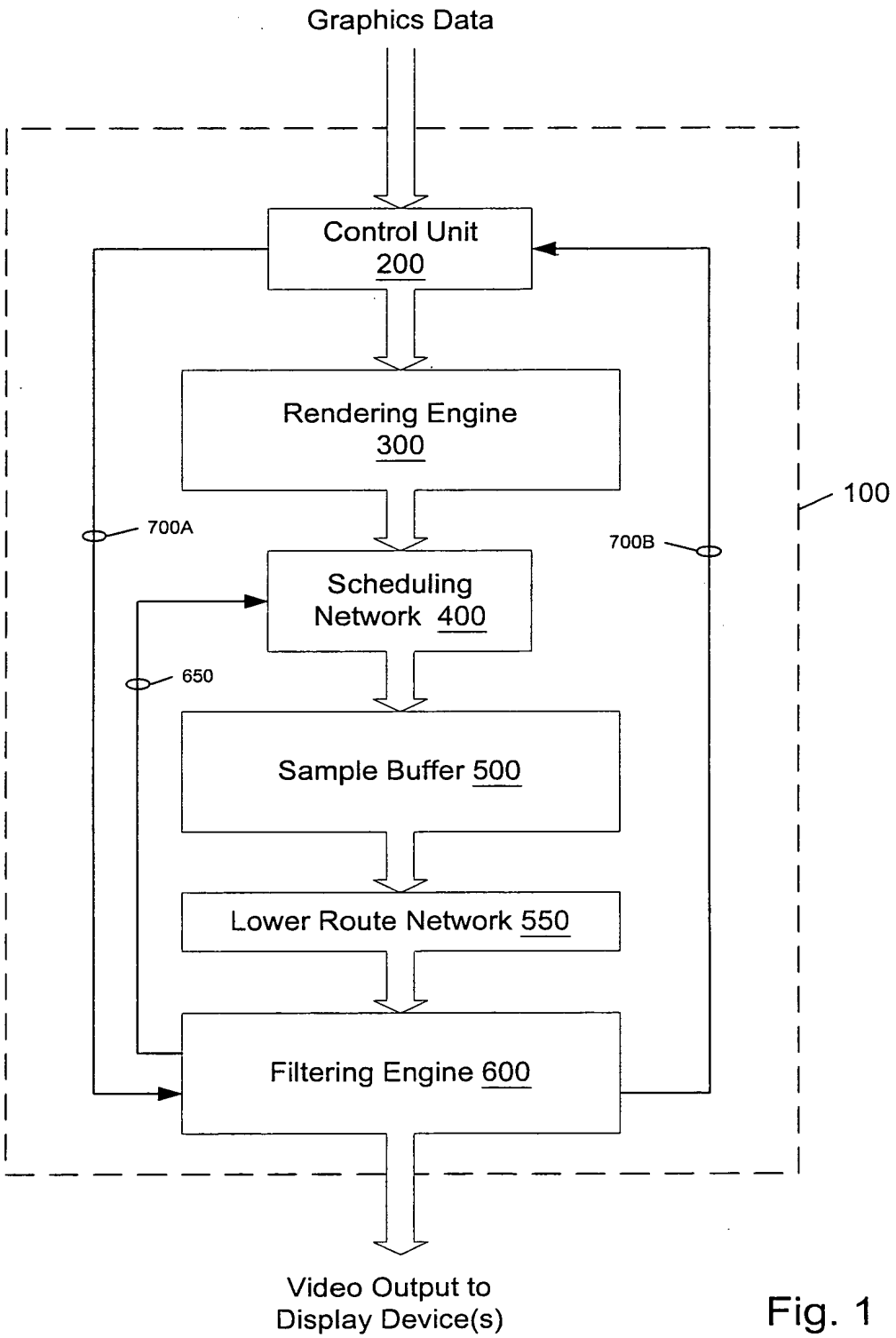

Graphics Data
Control Unit
200
Rendering Engine
300
100
700A
700B
Scheduling
Network 400
650
Sample Buffer 500
Lower Route Network 550
Filtering Engine 600
Video Output to
Display Device(s)

Fig. 1

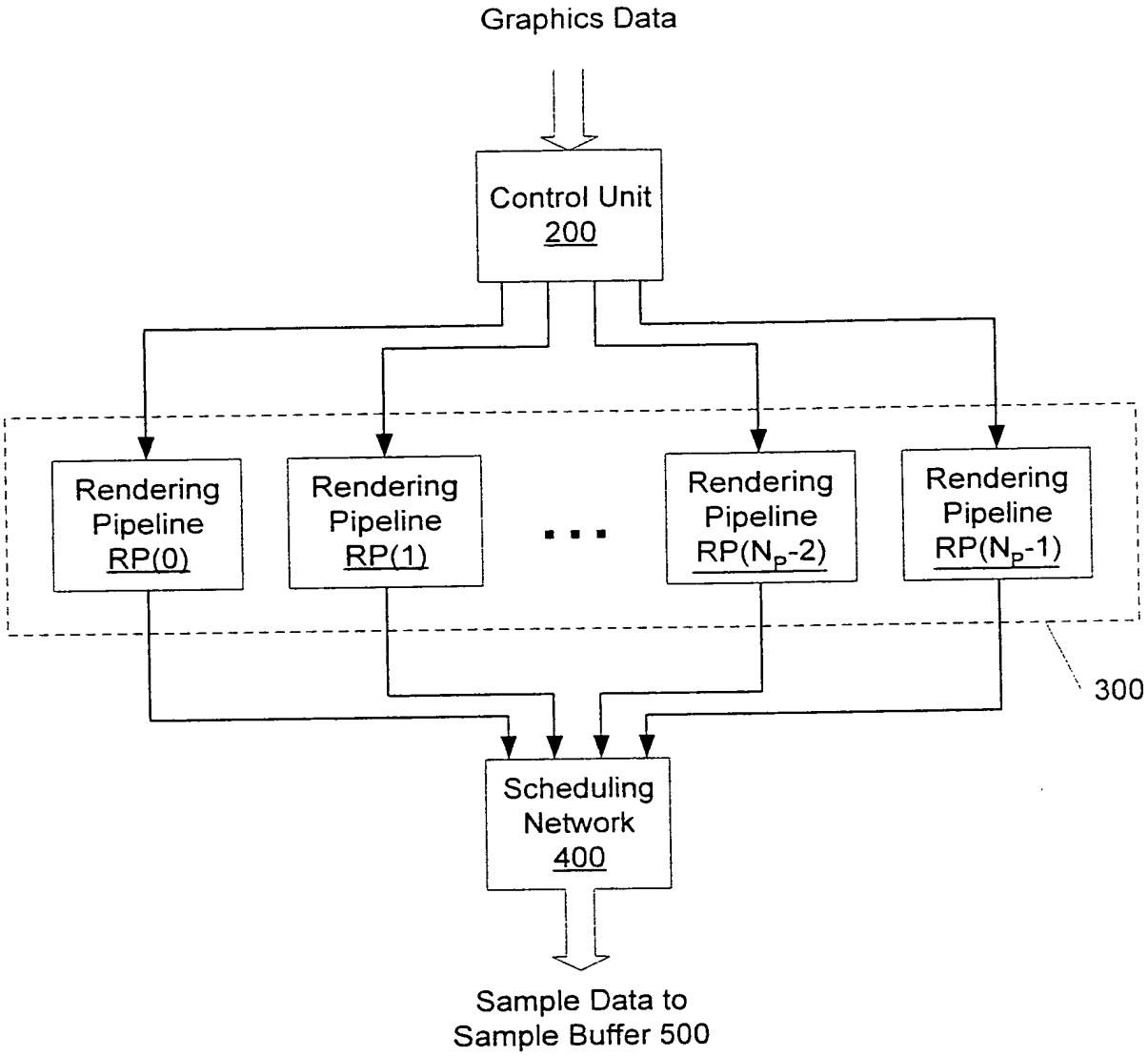

Graphics Data
Control Unit
200
Rendering
Pipeline
RP(0)
Rendering
Pipeline
RP(1)
Rendering
Pipeline
RP(N_P-2)
Rendering
Pipeline
RP(N_P-1)
300
Scheduling
Network
400
Sample Data to
Sample Buffer 500

Fig. 2

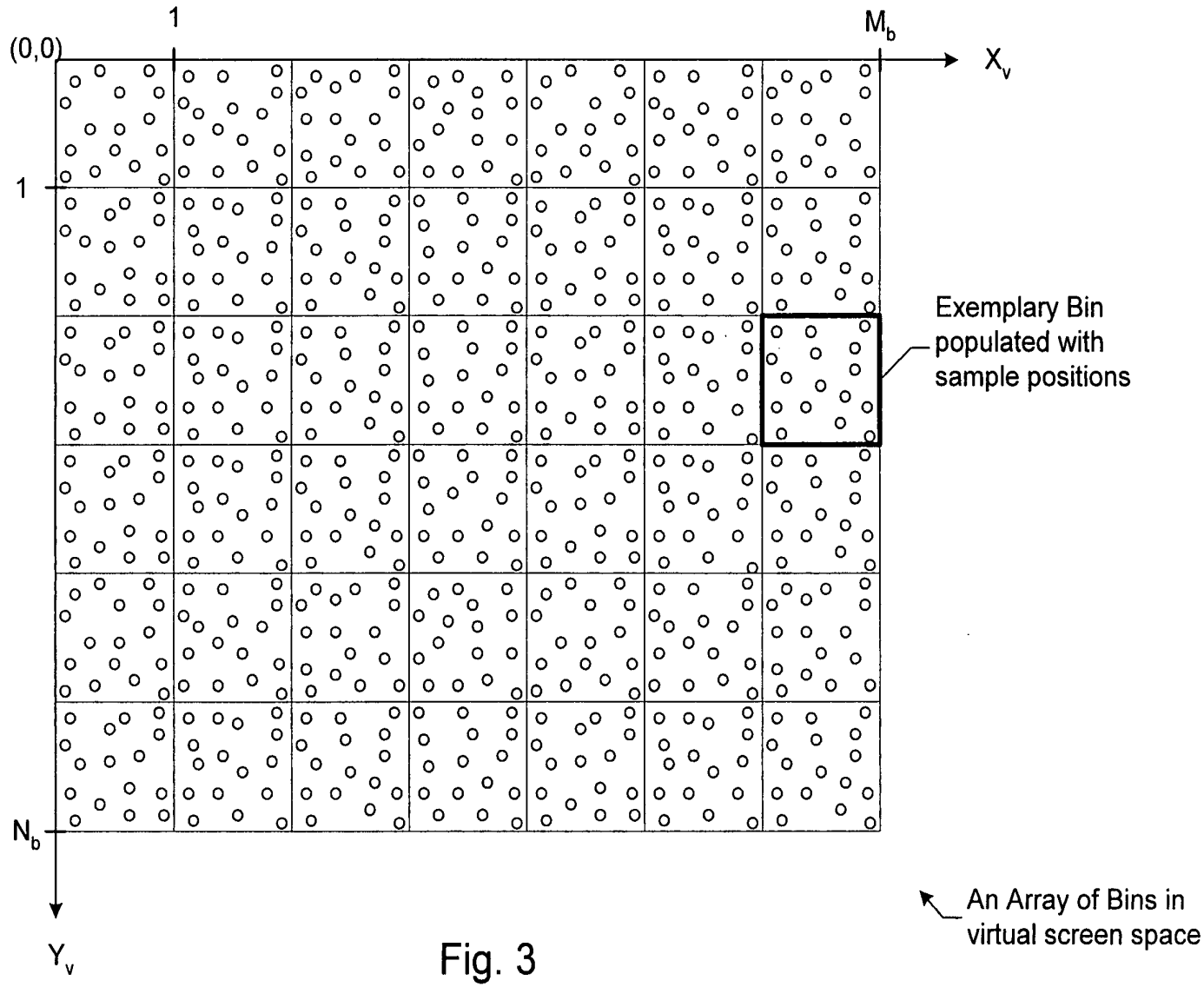

(0,0)
1
M_b
X_v
1
Exemplary Bin
populated with
sample positions
N_b
Y_v
An Array of Bins in
virtual screen space

Fig. 3

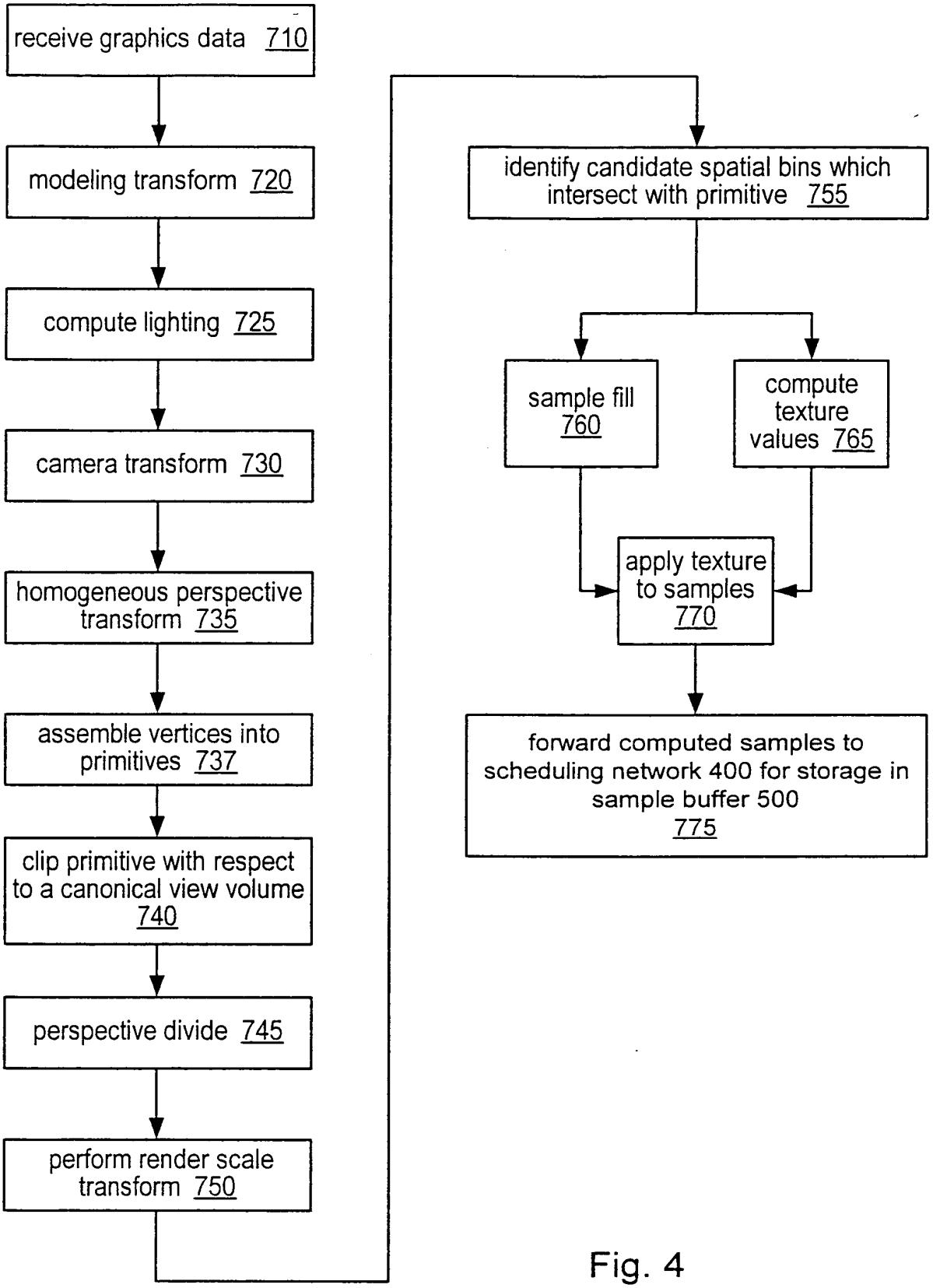

receive graphics data 710
modeling transform 720
compute lighting 725
camera transform 730
homogeneous perspective transform 735
assemble vertices into primitives 737
clip primitive with respect to a canonical view volume 740
perspective divide 745
perform render scale transform 750
identify candidate spatial bins which intersect with primitive 755
sample fill 760
compute texture values 765
apply texture to samples 770
forward computed samples to scheduling network 400 for storage in sample buffer 500 775

Fig. 4

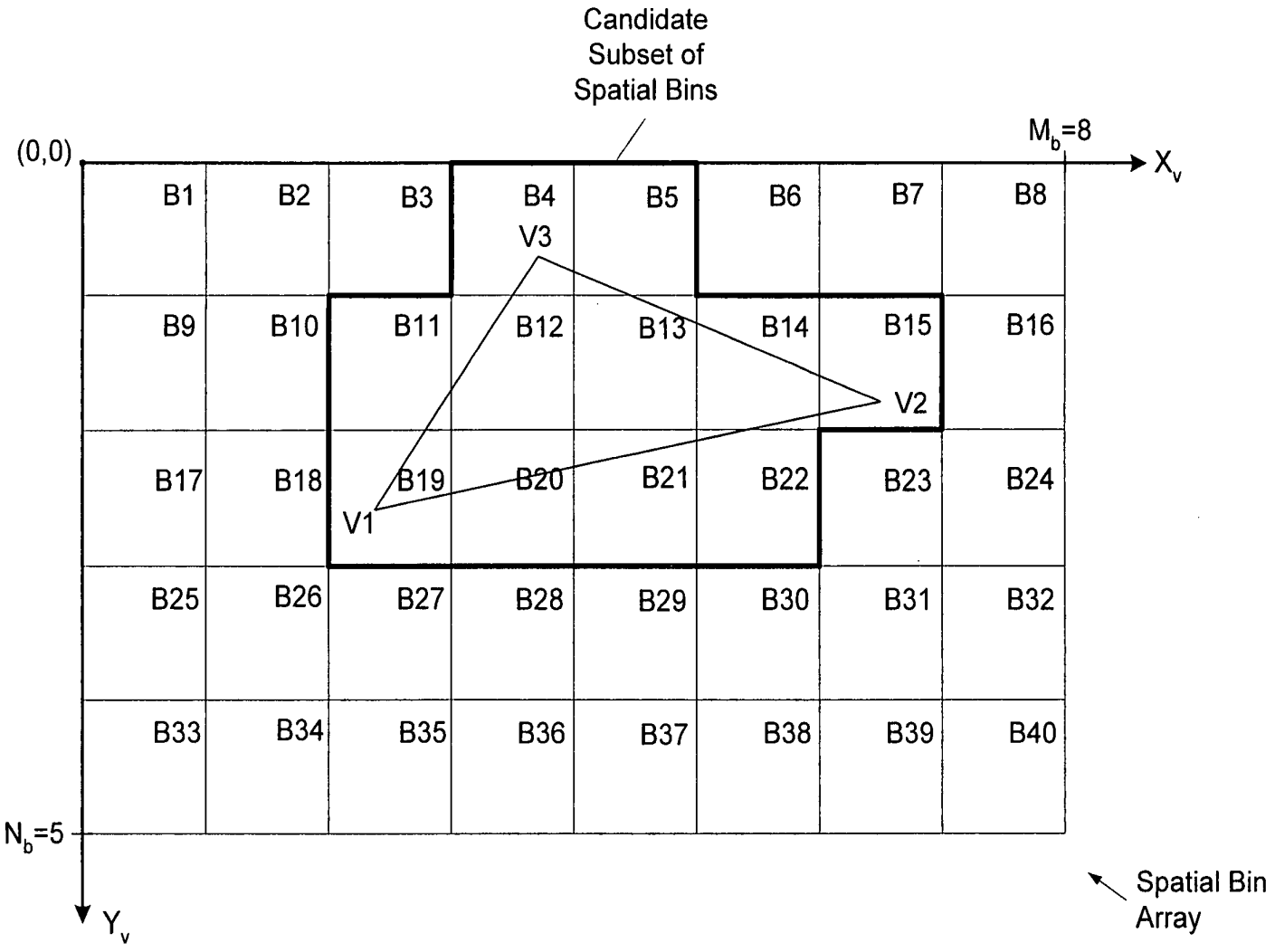

Candidate
Subset of
Spatial Bins
(0,0)
M_b=8
X_v
B1
B2
B3
B4
B5
B6
B7
B8
V3
B9
B10
B11
B12
B13
B14
B15
B16
V2
B17
B18
B19
B20
B21
B22
B23
B24
V1
B25
B26
B27
B28
B29
B30
B31
B32
B33
B34
B35
B36
B37
B38
B39
B40
N_b=5
Y_v
Spatial Bin
Array

Fig. 5

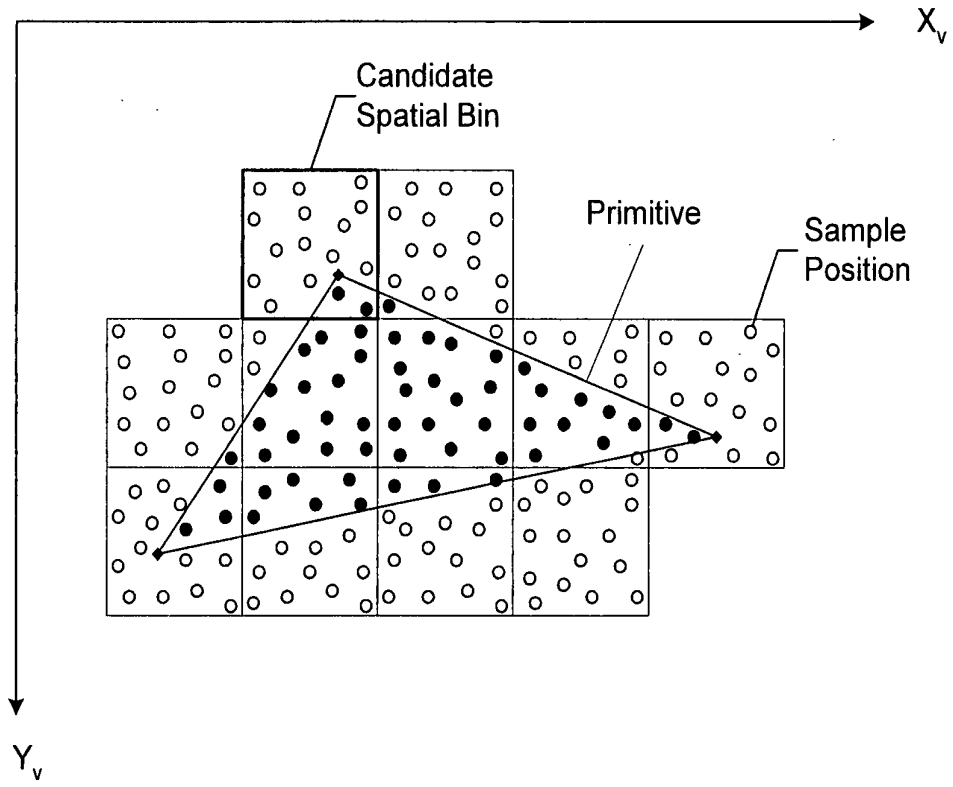


Fig. 6

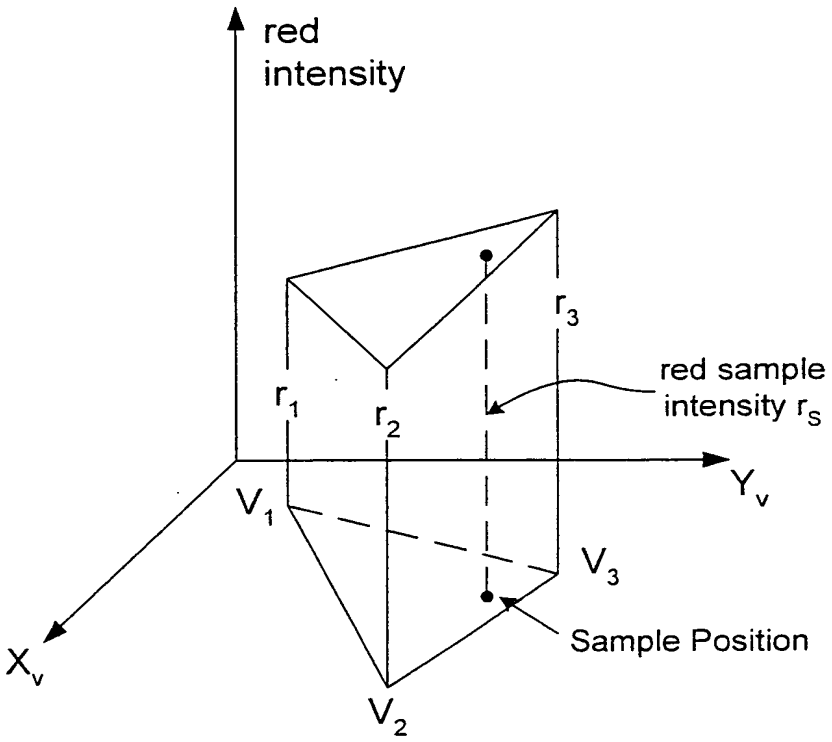

red
intensity
r3
red sample
intensity rS
r1
r2
Yv
V1
V3
Sample Position
Xv
V2

Fig. 7

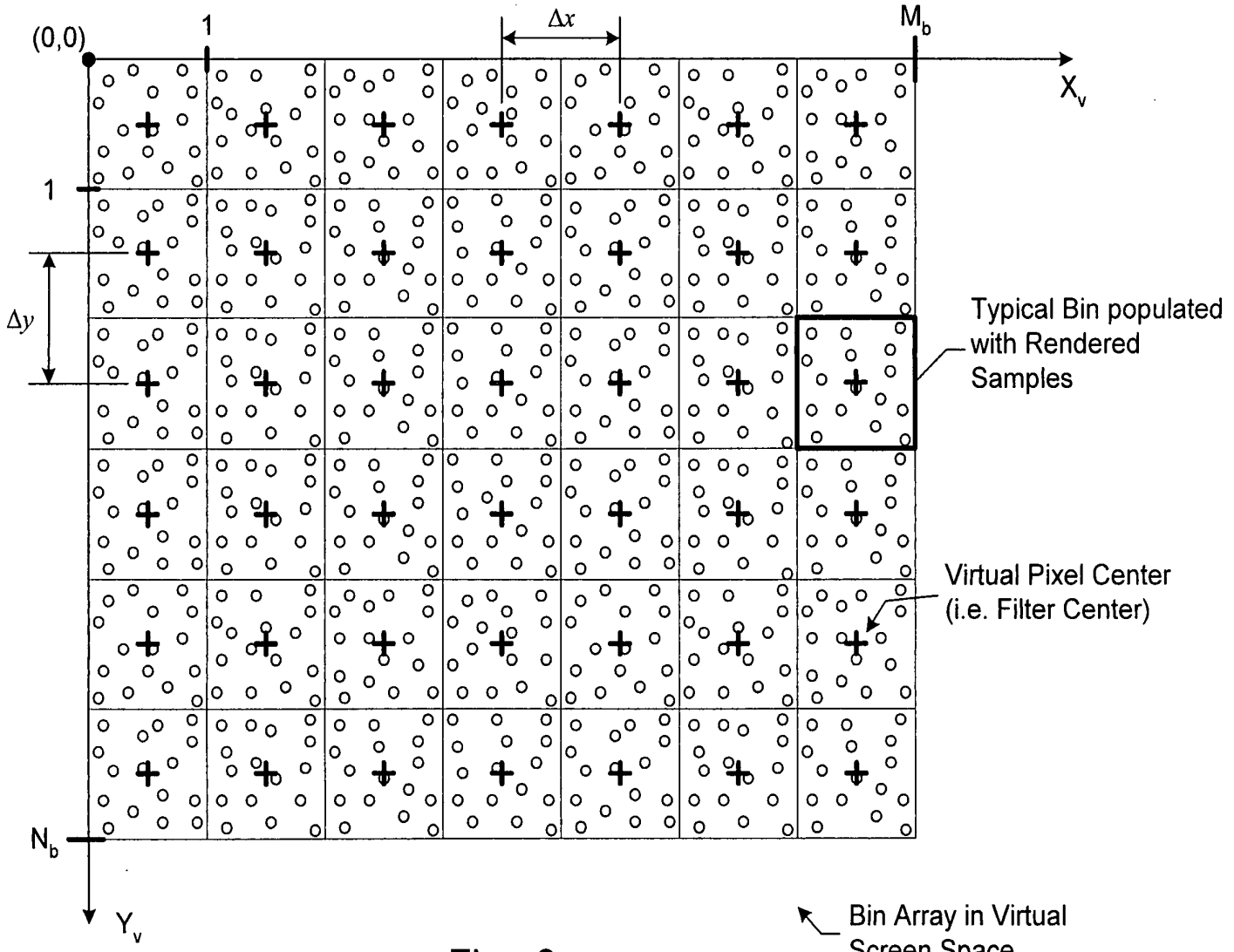

(0,0)
1
Δx
M_b
X_v
1
Δy
Typical Bin populated with Rendered Samples
Virtual Pixel Center (i.e. Filter Center)
N_b
Y_v
Bin Array in Virtual Screen Space

Fig. 8

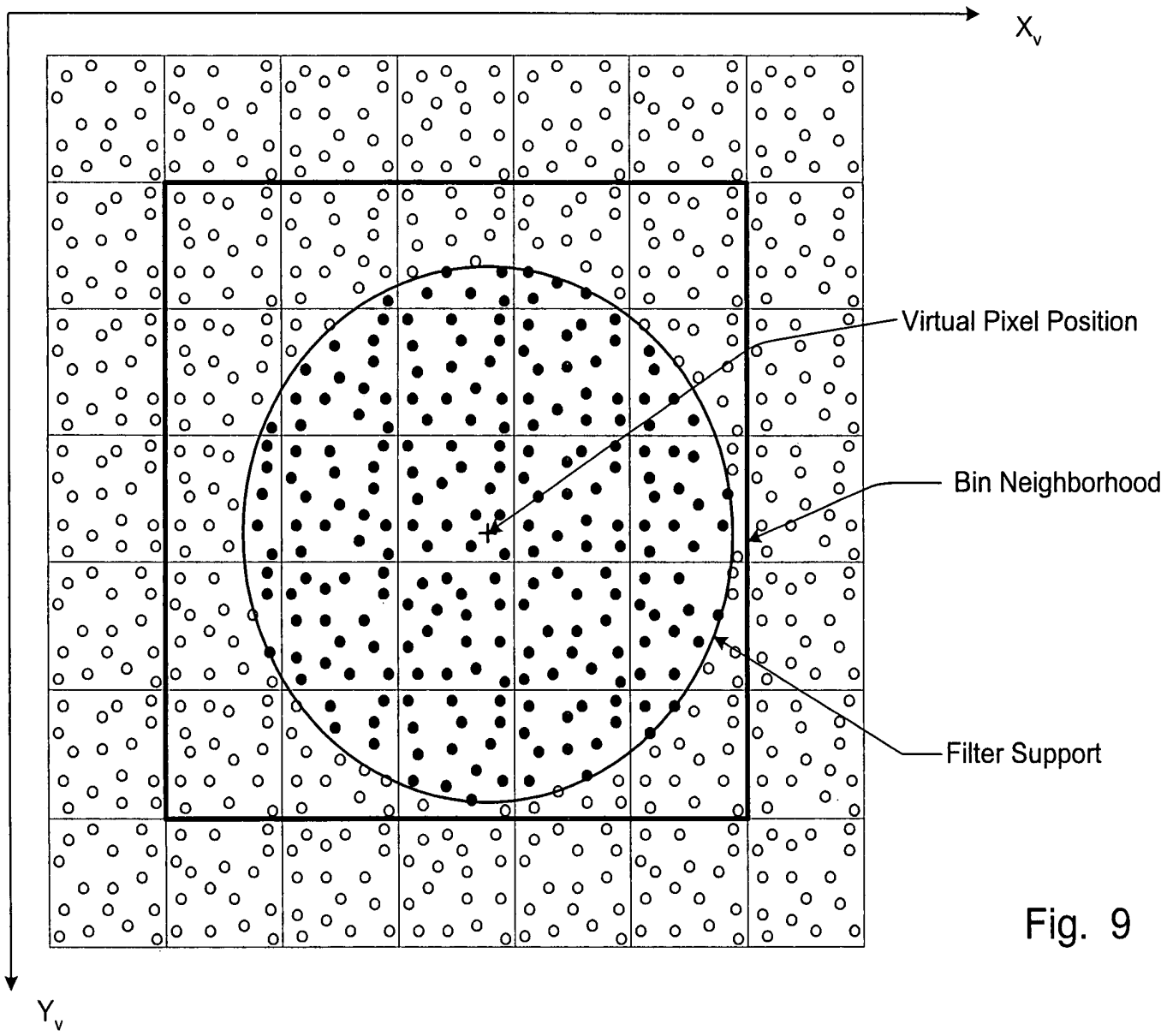

X_v
Y_v
Virtual Pixel Position
Bin Neighborhood
Filter Support

Fig. 9

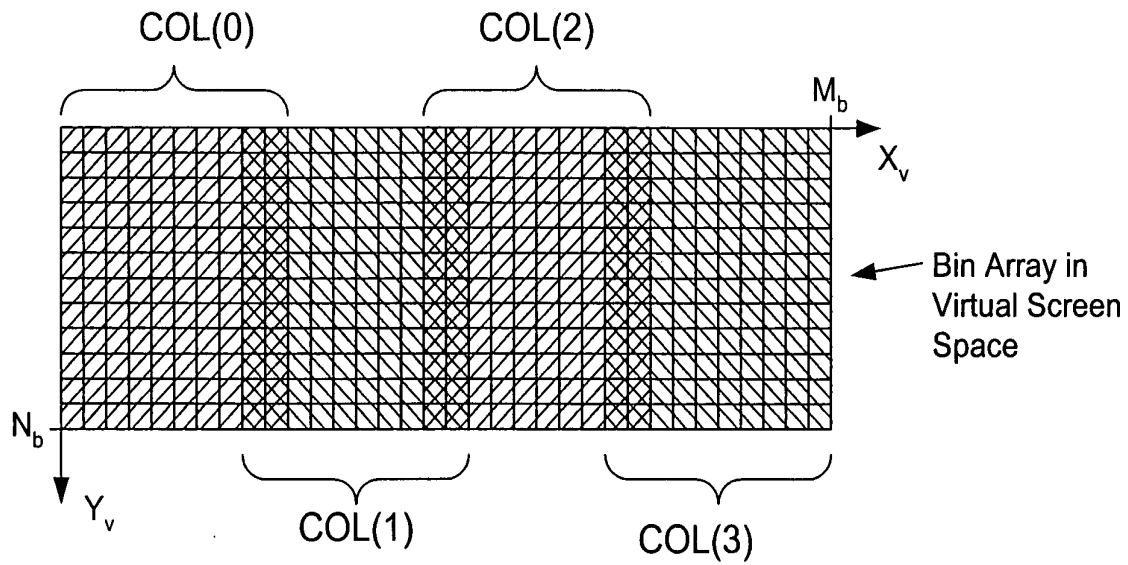

COL(0)
COL(2)
M_b
X_v
Bin Array in Virtual Screen Space
N_b
Y_v
COL(1)
COL(3)

FIG. 10

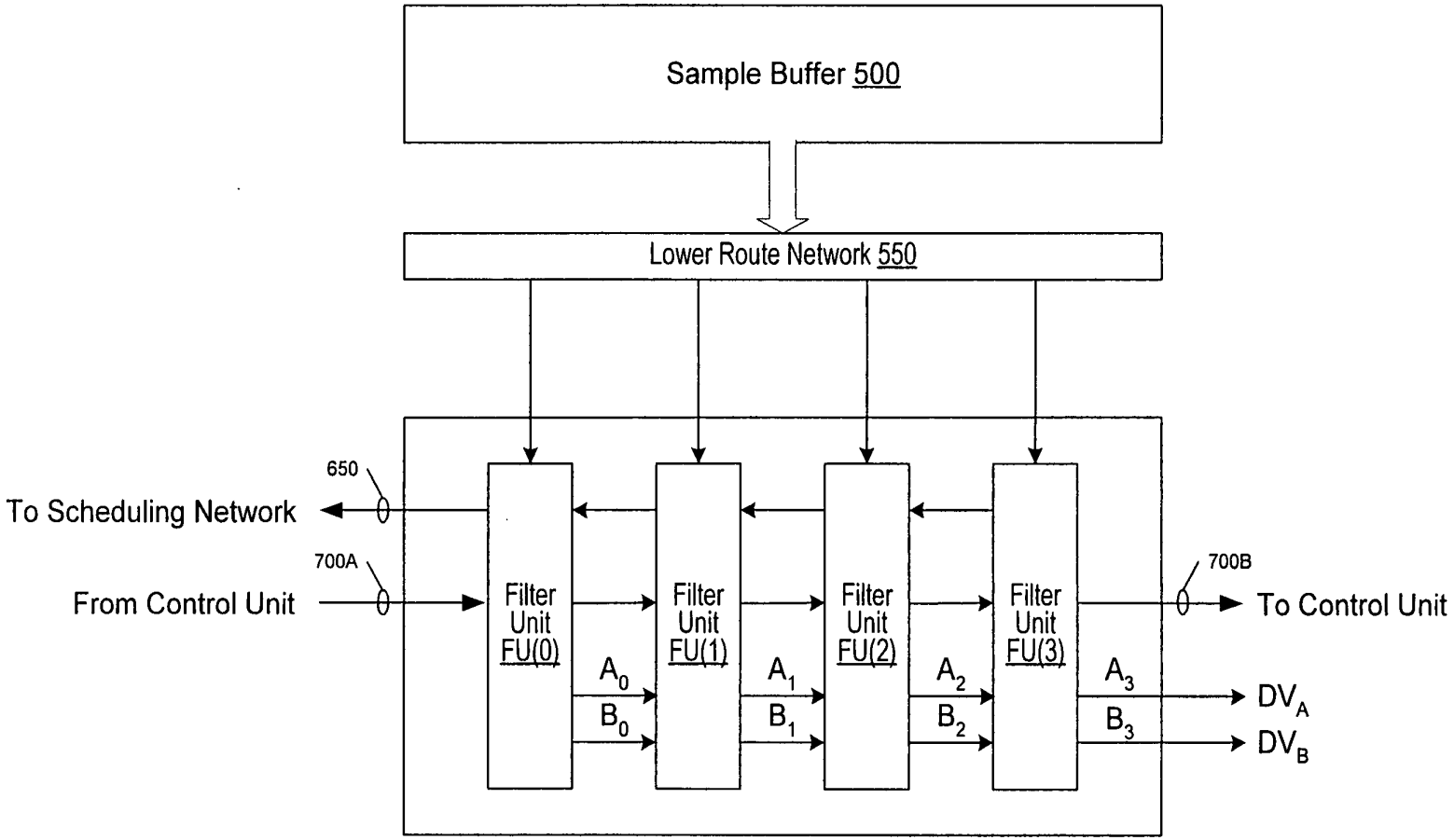

Sample Buffer 500
Lower Route Network 550
650
To Scheduling Network
700A
From Control Unit
Filter Unit FU(0)
Filter Unit FU(1)
Filter Unit FU(2)
Filter Unit FU(3)
A0
B0
A1
B1
A2
B2
A3
B3
700B
To Control Unit
DVA
DVB

FIG. 11

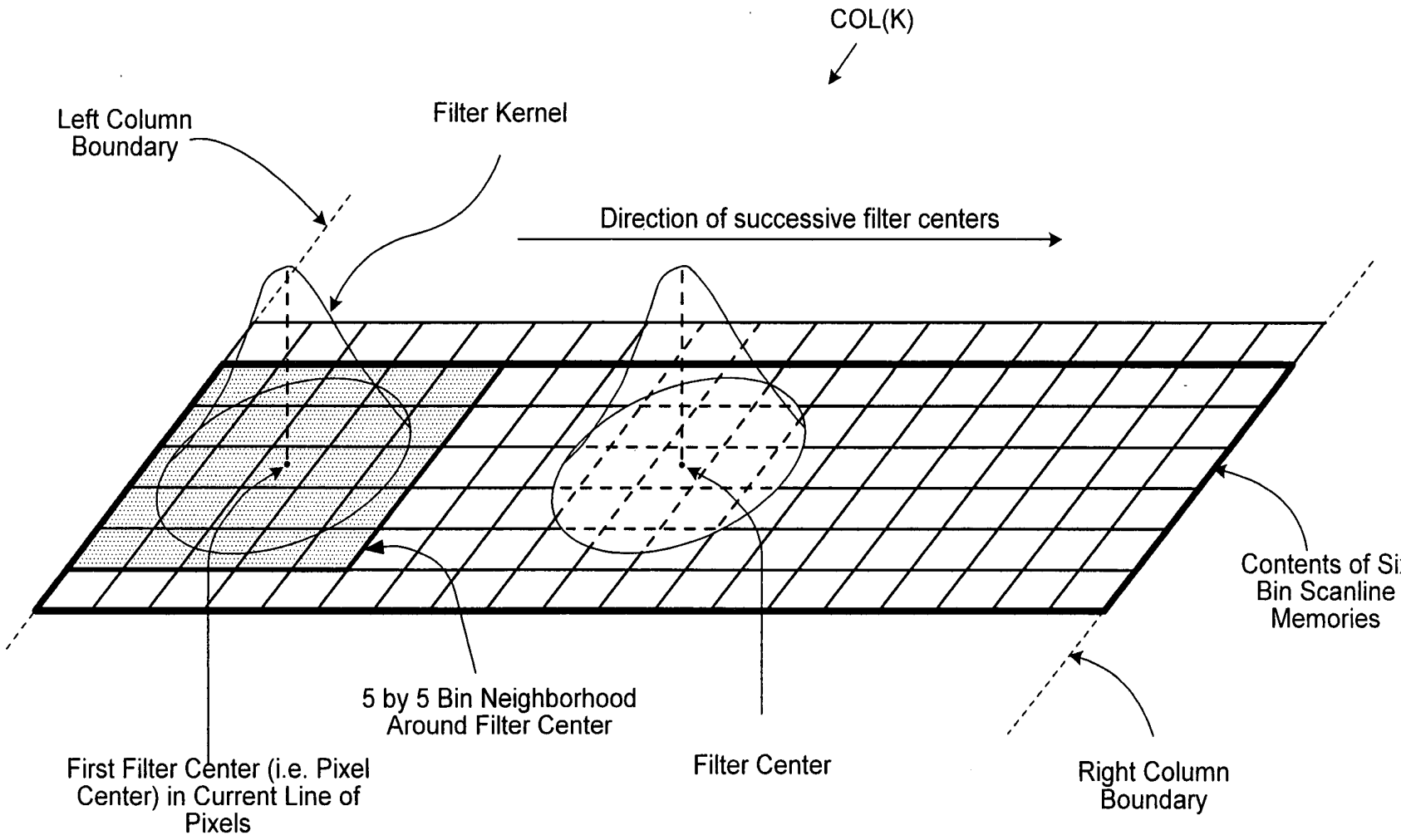

COL(K)
Left Column Boundary
Filter Kernel
Direction of successive filter centers
Contents of Six Bin Scanline Memories
5 by 5 Bin Neighborhood Around Filter Center
First Filter Center (i.e. Pixel Center) in Current Line of Pixels
Filter Center
Right Column Boundary

FIG. 12

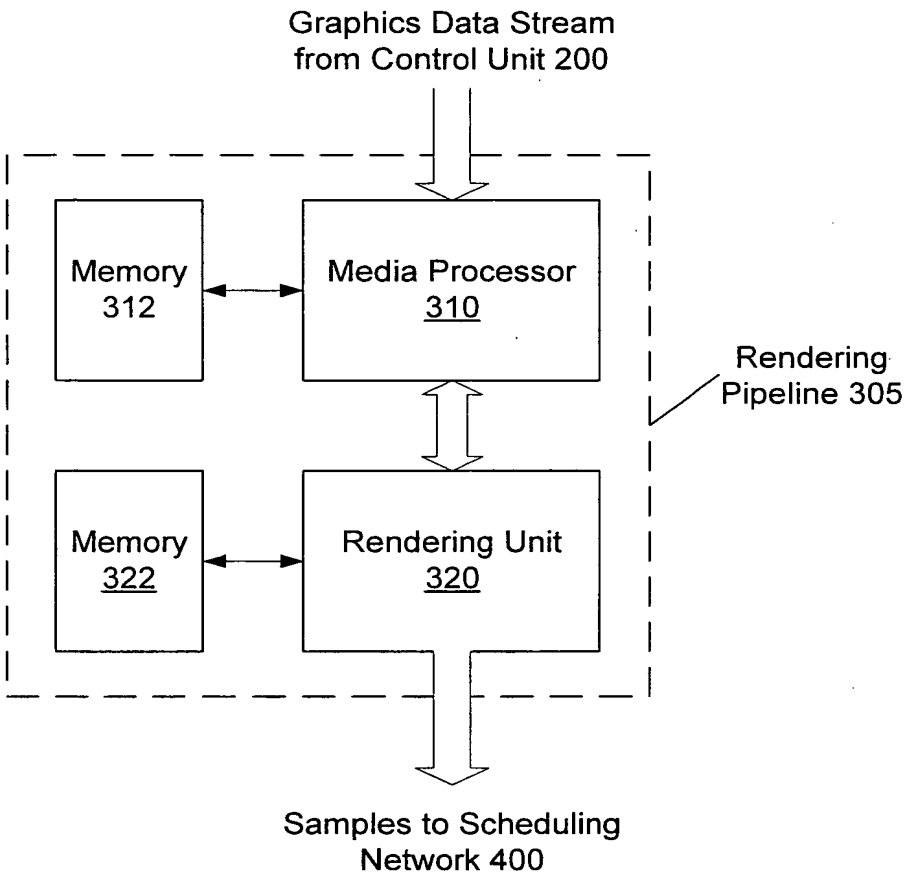

Graphics Data Stream
from Control Unit 200
Memory
312
Media Processor
310
Rendering
Pipeline 305
Memory
322
Rendering Unit
320
Samples to Scheduling
Network 400

Fig. 13

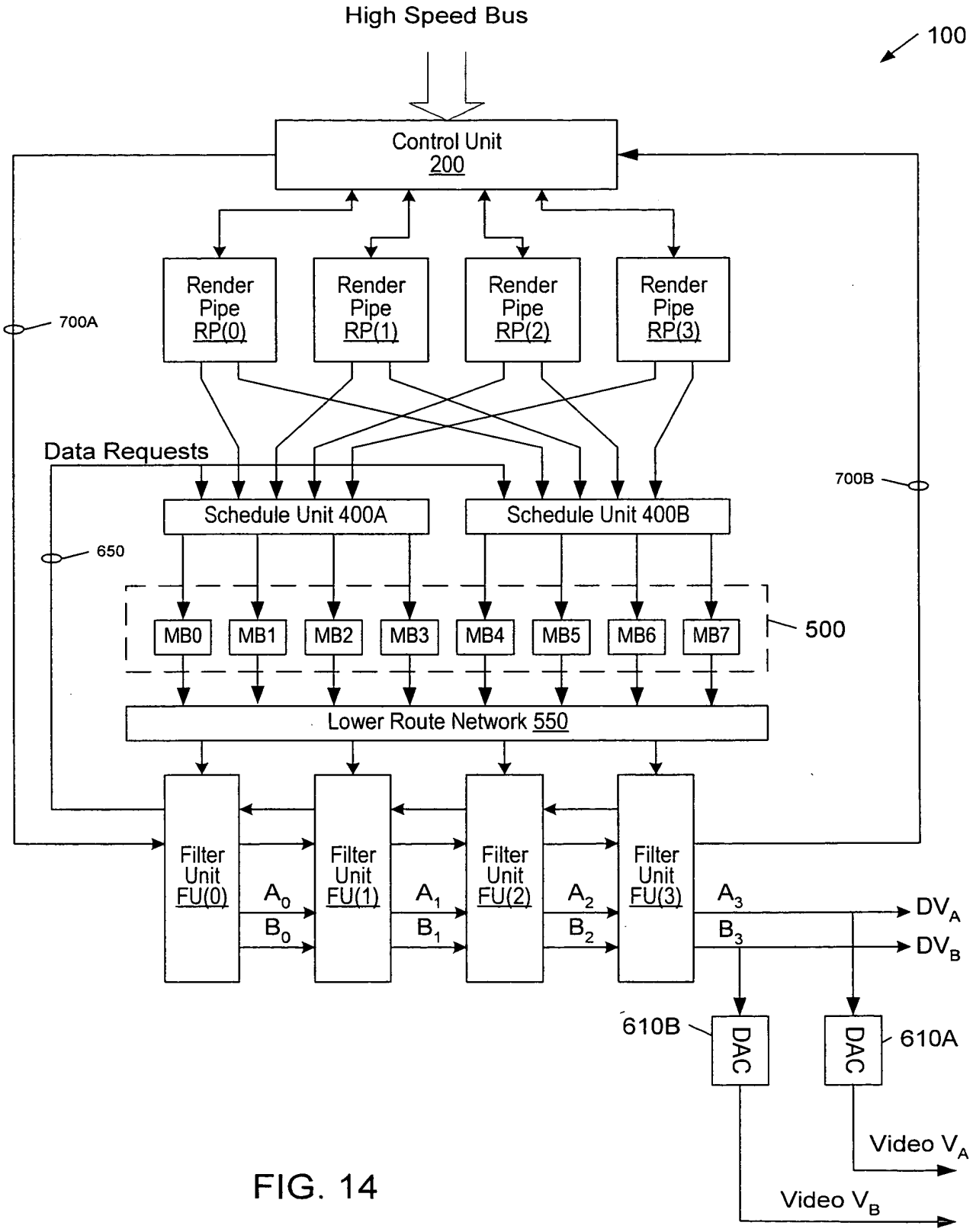

High Speed Bus
100
Control Unit
200
Render Pipe RP(0)
Render Pipe RP(1)
Render Pipe RP(2)
Render Pipe RP(3)
700A
Data Requests
700B
Schedule Unit 400A
Schedule Unit 400B
650
MB0
MB1
MB2
MB3
MB4
MB5
MB6
MB7
500
Lower Route Network 550
Filter Unit FU(0)
Filter Unit FU(1)
Filter Unit FU(2)
Filter Unit FU(3)
A0
B0
A1
B1
A2
B2
A3
B3
DVA
DVB
610B
DAC
DAC
610A
Video VA
Video VB

FIG. 14

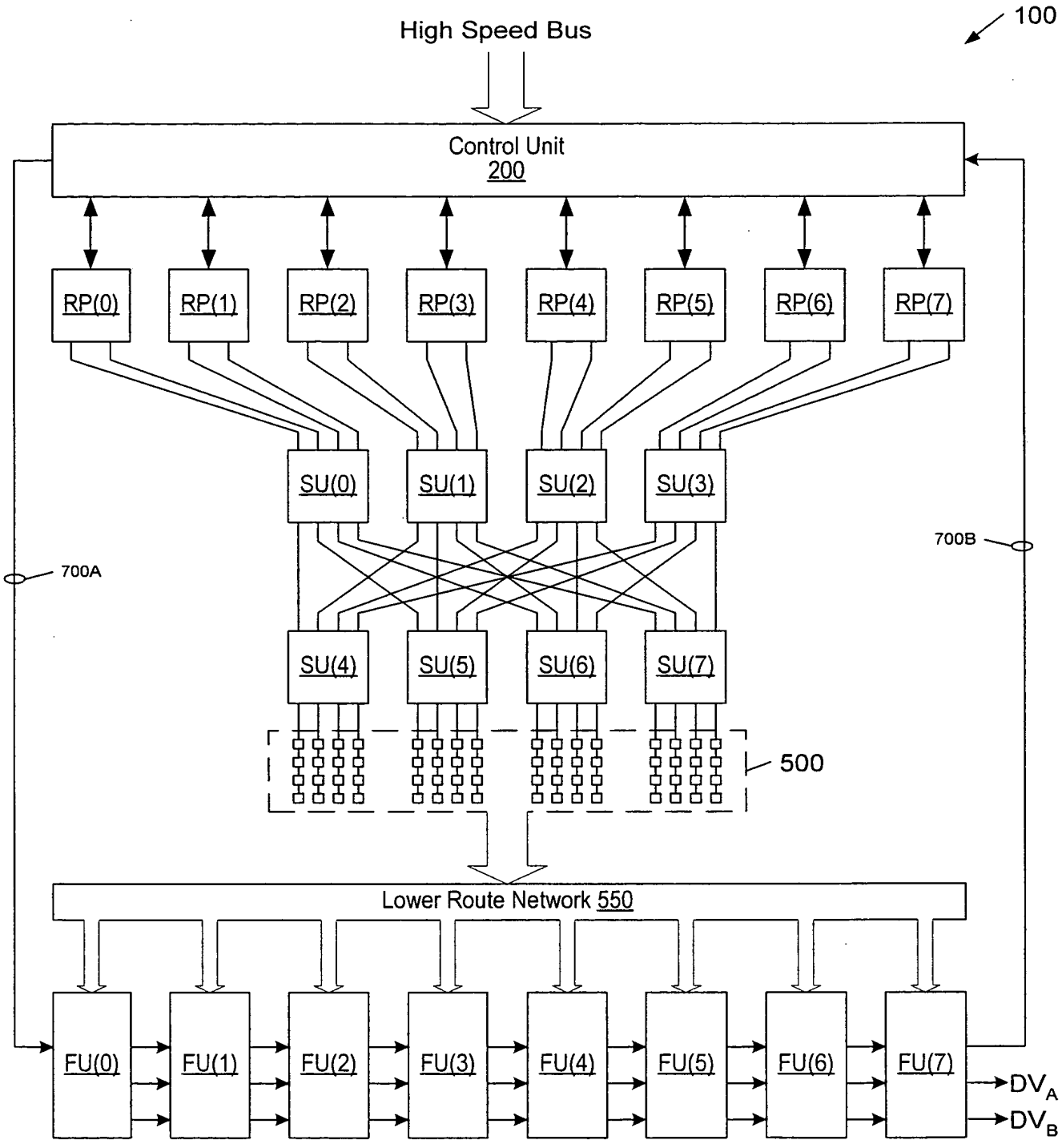

100
High Speed Bus
Control Unit
200
RP(0)
RP(1)
RP(2)
RP(3)
RP(4)
RP(5)
RP(6)
RP(7)
SU(0)
SU(1)
SU(2)
SU(3)
SU(4)
SU(5)
SU(6)
SU(7)
700A
700B
500
Lower Route Network 550
FU(0)
FU(1)
FU(2)
FU(3)
FU(4)
FU(5)
FU(6)
FU(7)
DV_A
DV_B

Fig. 15

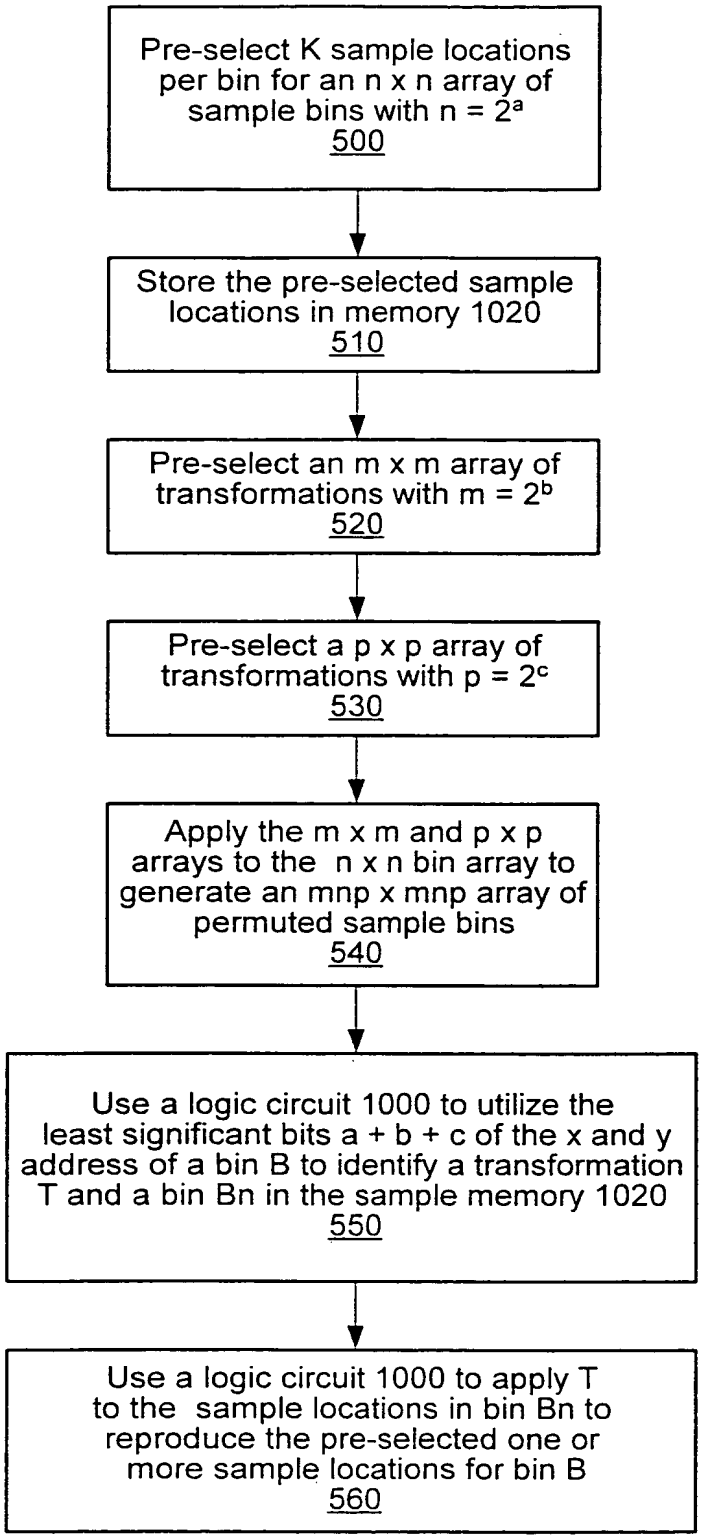
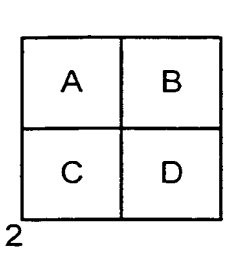

Pre-select K sample locations per bin for an n x n array of sample bins with n = 2^a
500
Store the pre-selected sample locations in memory 1020
510
Pre-select an m x m array of transformations with m = 2^b
520
Pre-select a p x p array of transformations with p = 2^c
530
Apply the m x m and p x p arrays to the n x n bin array to generate an mnp x mnp array of permuted sample bins
540
Use a logic circuit 1000 to utilize the least significant bits a + b + c of the x and y address of a bin B to identify a transformation T and a bin Bn in the sample memory 1020
550
Use a logic circuit 1000 to apply T to the sample locations in bin Bn to reproduce the pre-selected one or more sample locations for bin B
560

Fig. 16

Populate a 2 x 2 Sample Bin Array



1 to 16 Sample Locations / Bin

Fig. 17a

Select an Array of Transformations from:

iy	ix	s	Transformation
0	0	0	none
0	0	1	swapXY: mirror about y = -x -1
0	1	0	invertX: mirror about y
0	1	1	rotate 270 degrees clockwise
1	0	0	invertY: mirror about x
1	0	1	rotate 90 degrees clockwise
1	1	0	rotate 180 degrees
1	1	1	mirror about y = x

Fig. 17b

Inner Transformation Array
(xb,yd),(xc,yb),(xd,yc)

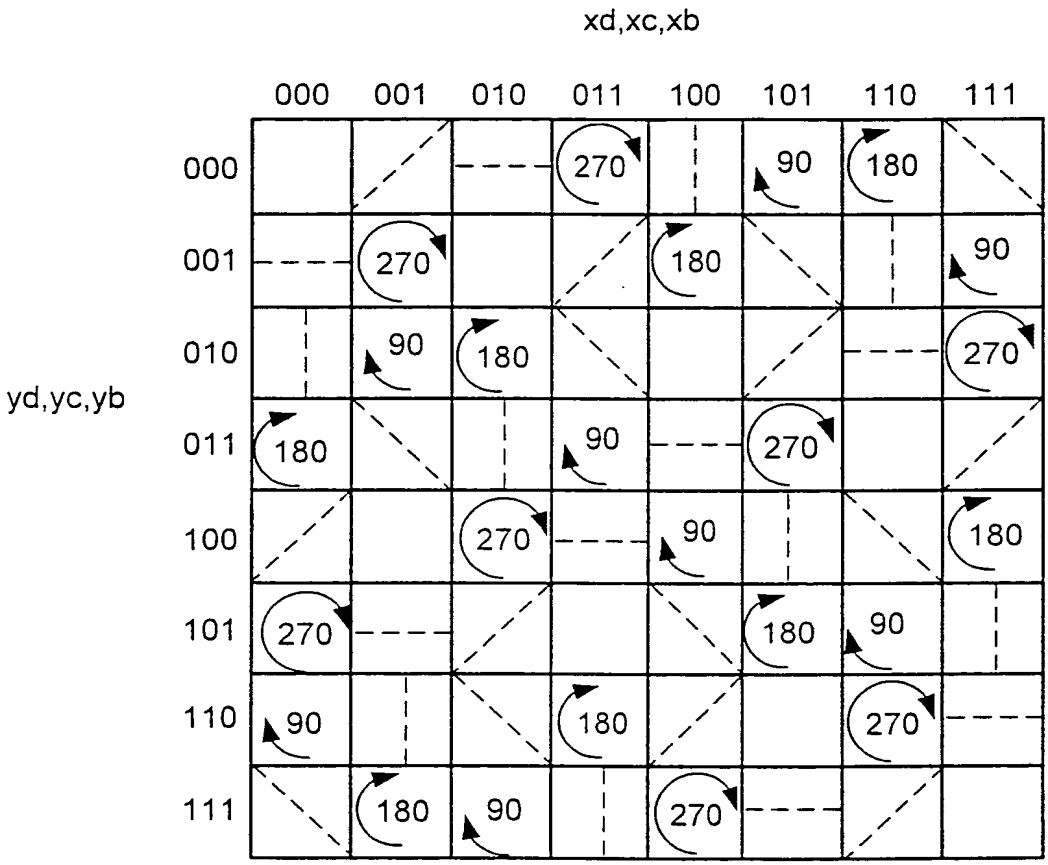


Fig. 18

Outer Transformation Array
(x4,y5),(x6,y4),(x5,y6)

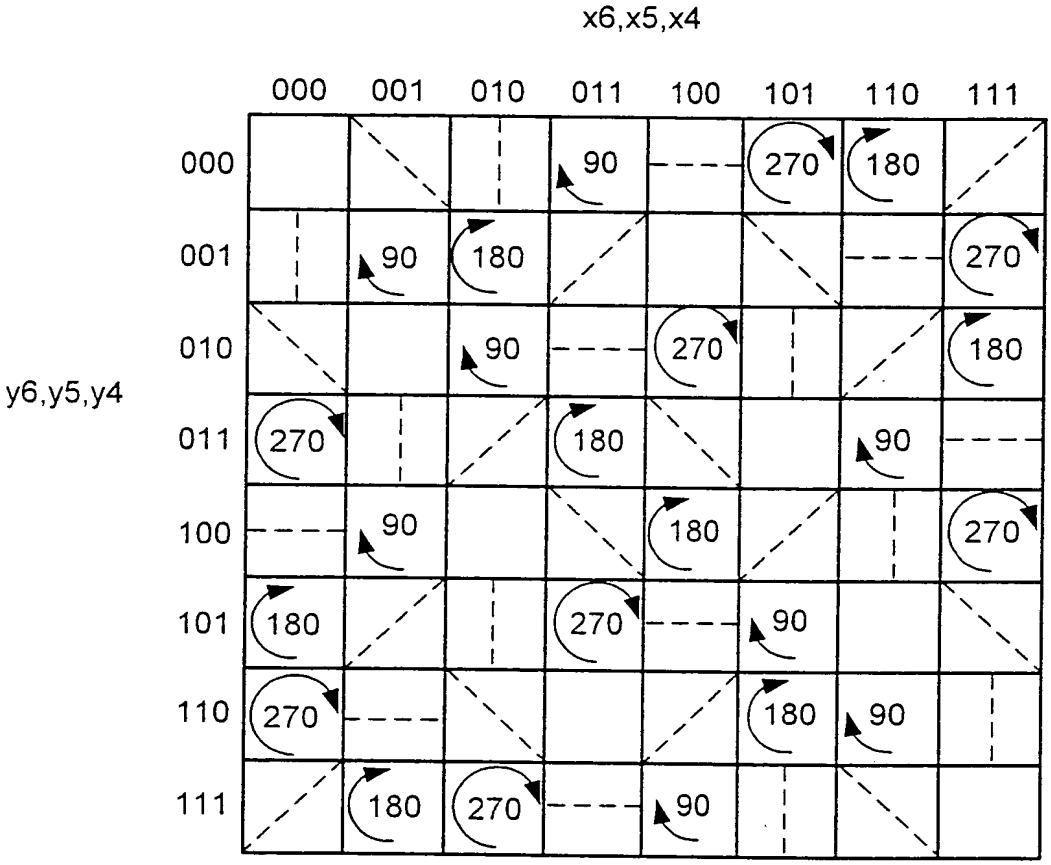


Fig. 19

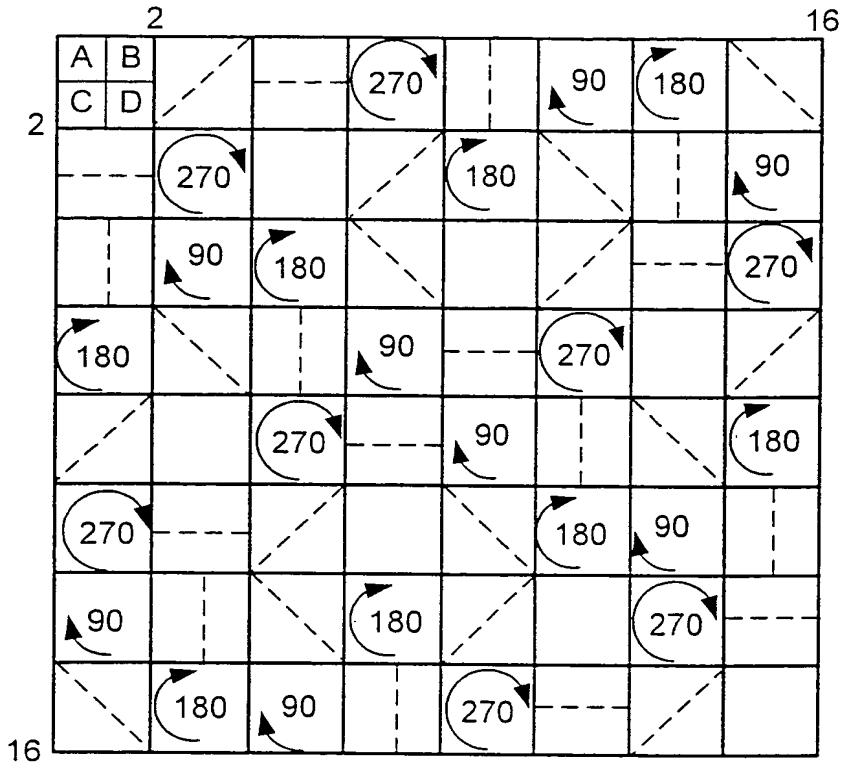


Fig. 20a

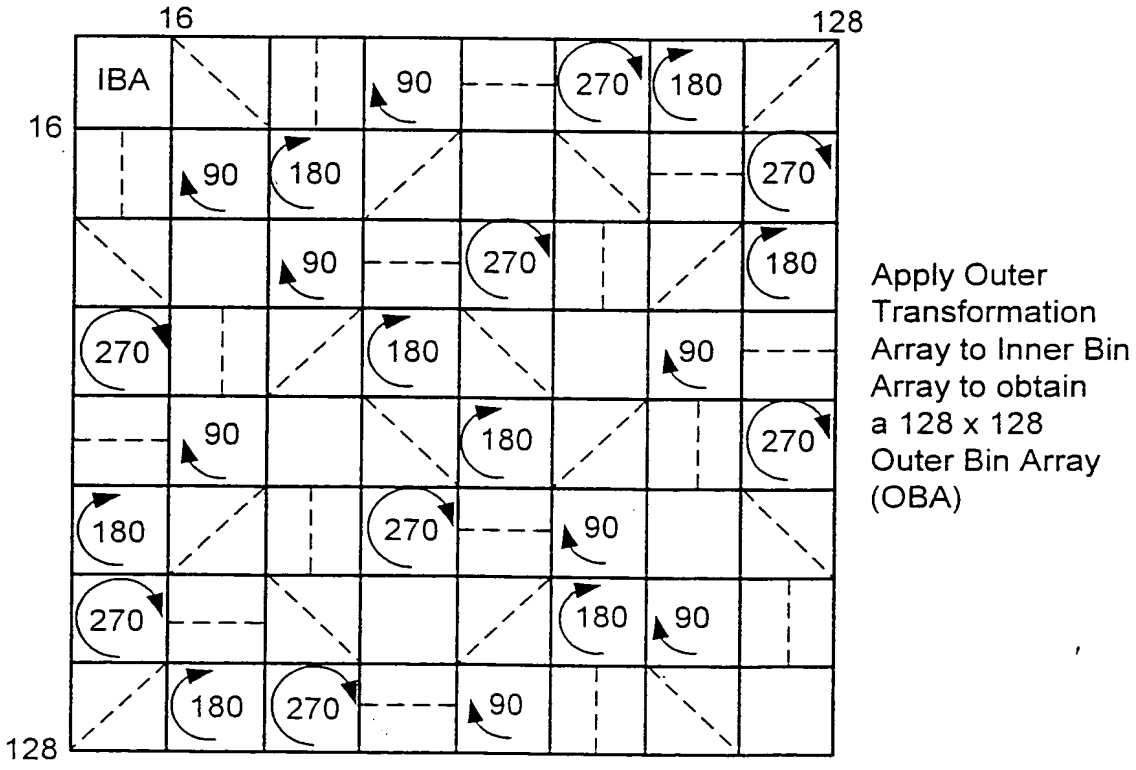


Fig. 20b

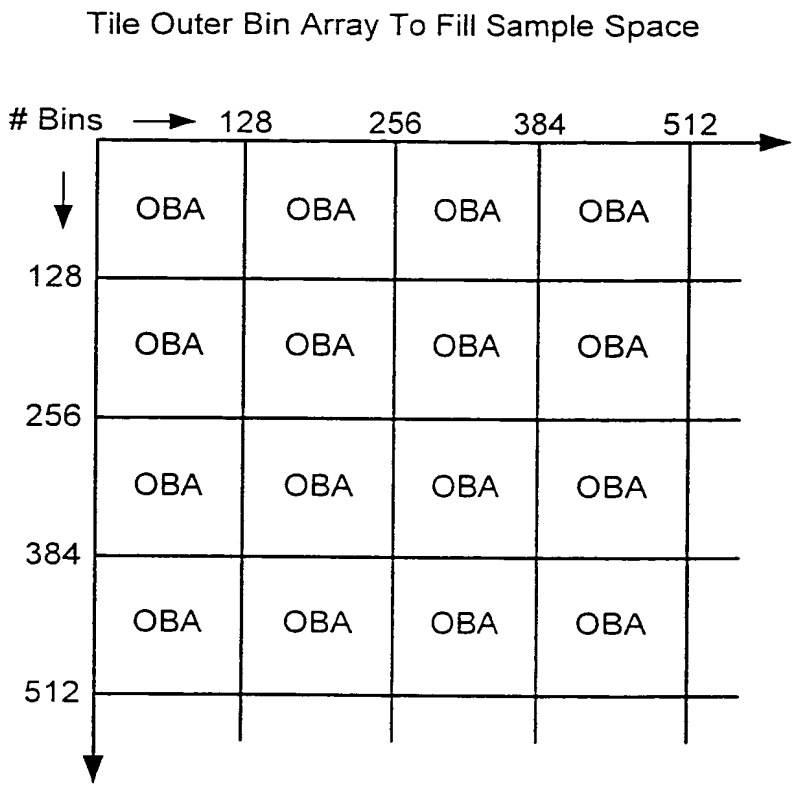

Tile Outer Bin Array To Fill Sample Space
Bins
128
256
384
512
OBA

Fig. 20c

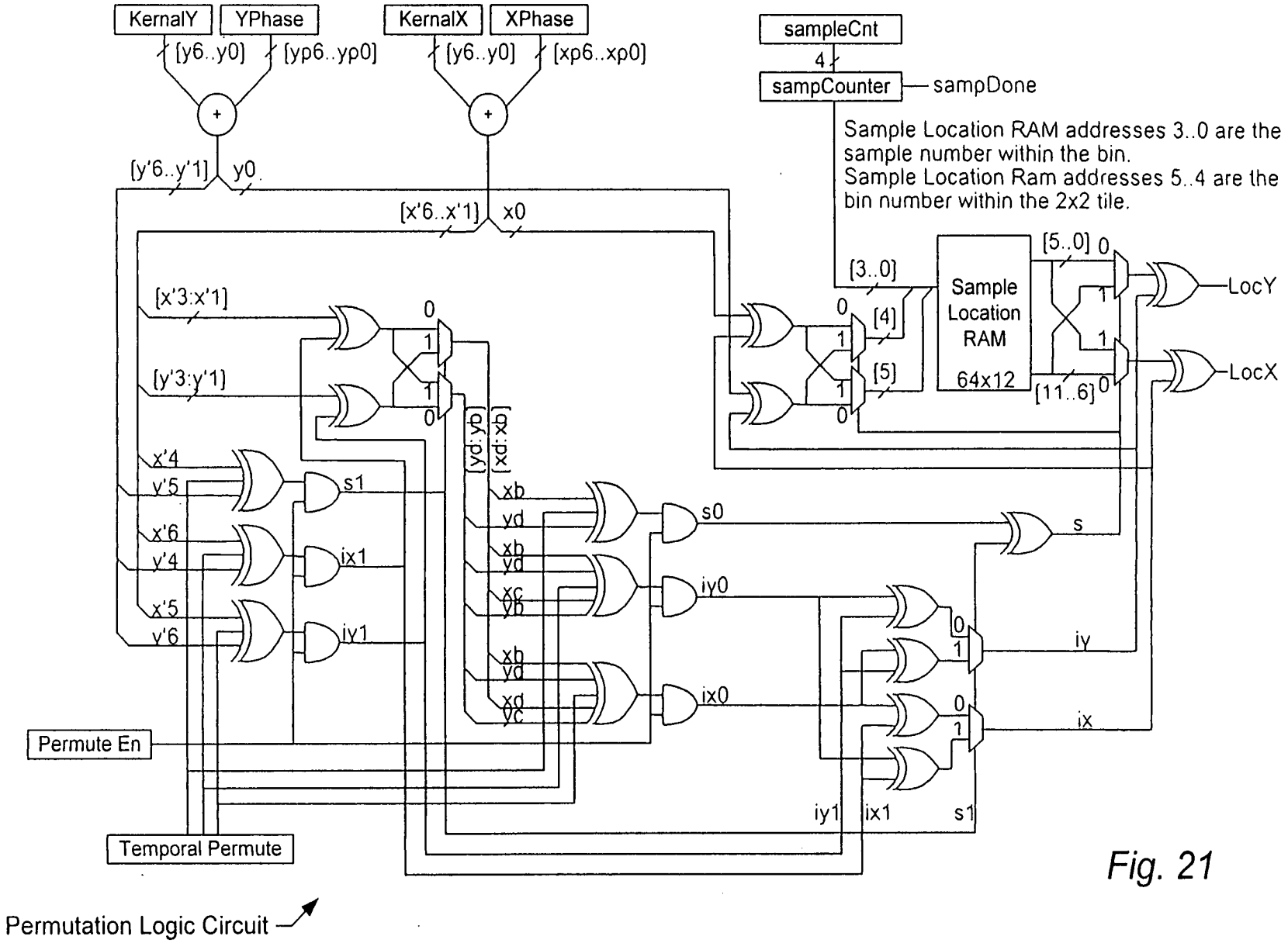

KernalY
YPhase
KernalX
XPhase
[y6..y0]
[yp6..yp0]
[y6..y0]
[xp6..xp0]
sampleCnt
4
sampCounter
sampDone
Sample Location RAM addresses 3..0 are the sample number within the bin.
Sample Location Ram addresses 5..4 are the bin number within the 2x2 tile.
[y'6..y'1]
y0
[x'6..x'1]
x0
[3..0]
Sample Location RAM
64x12
[5..0]
[11..6]
LocY
LocX
[x'3:x'1]
[y'3:y'1]
[yd:yb]
[xd:xb]
x'4
y'5
x'6
y'4
x'5
y'6
s1
ix1
iy1
xb
yd
xc
yc
xd
s0
iy0
ix0
s
iy
ix
Permute En
Temporal Permute
iy1
ix1
s1
Permutation Logic Circuit

Fig. 21

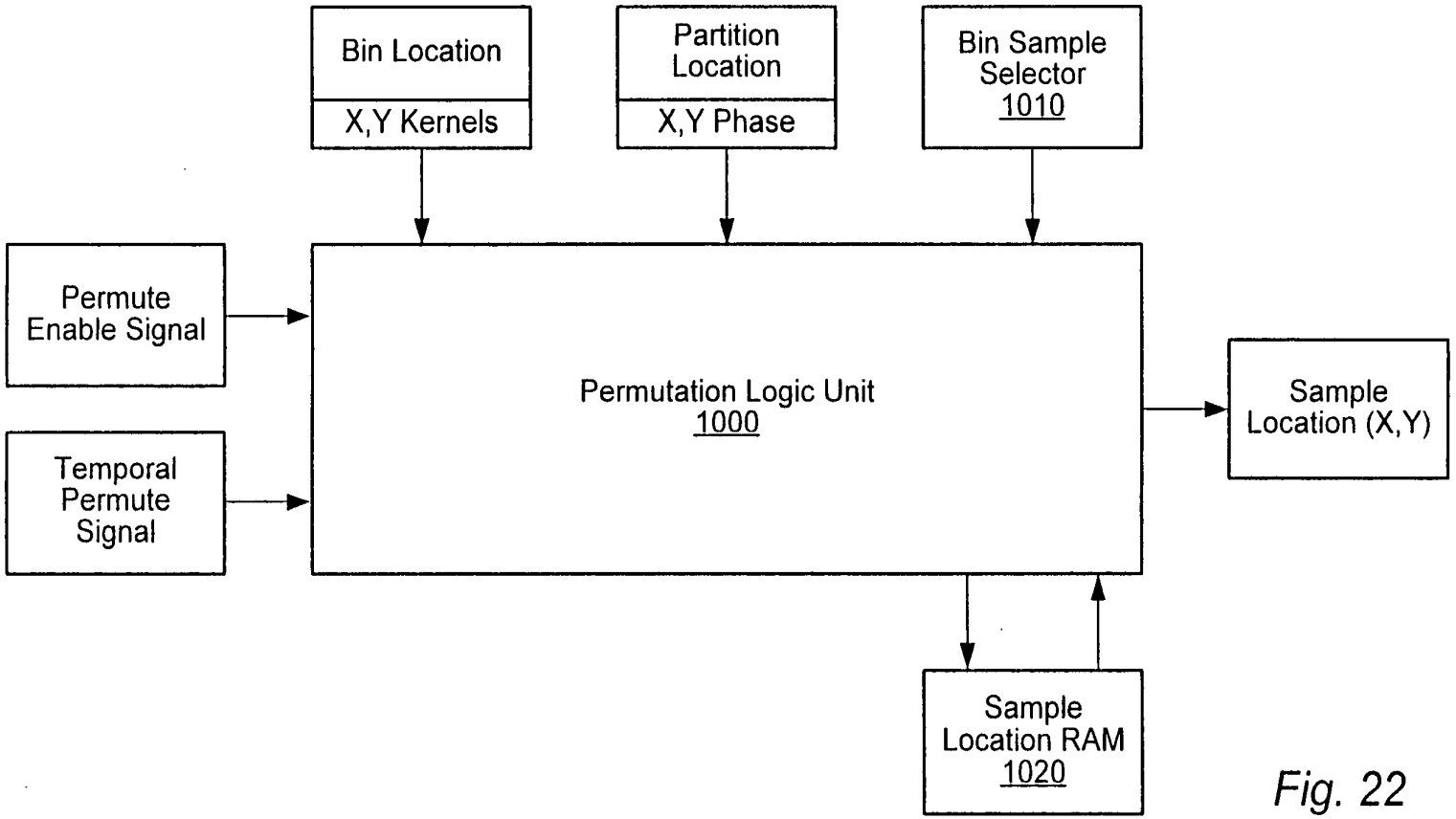
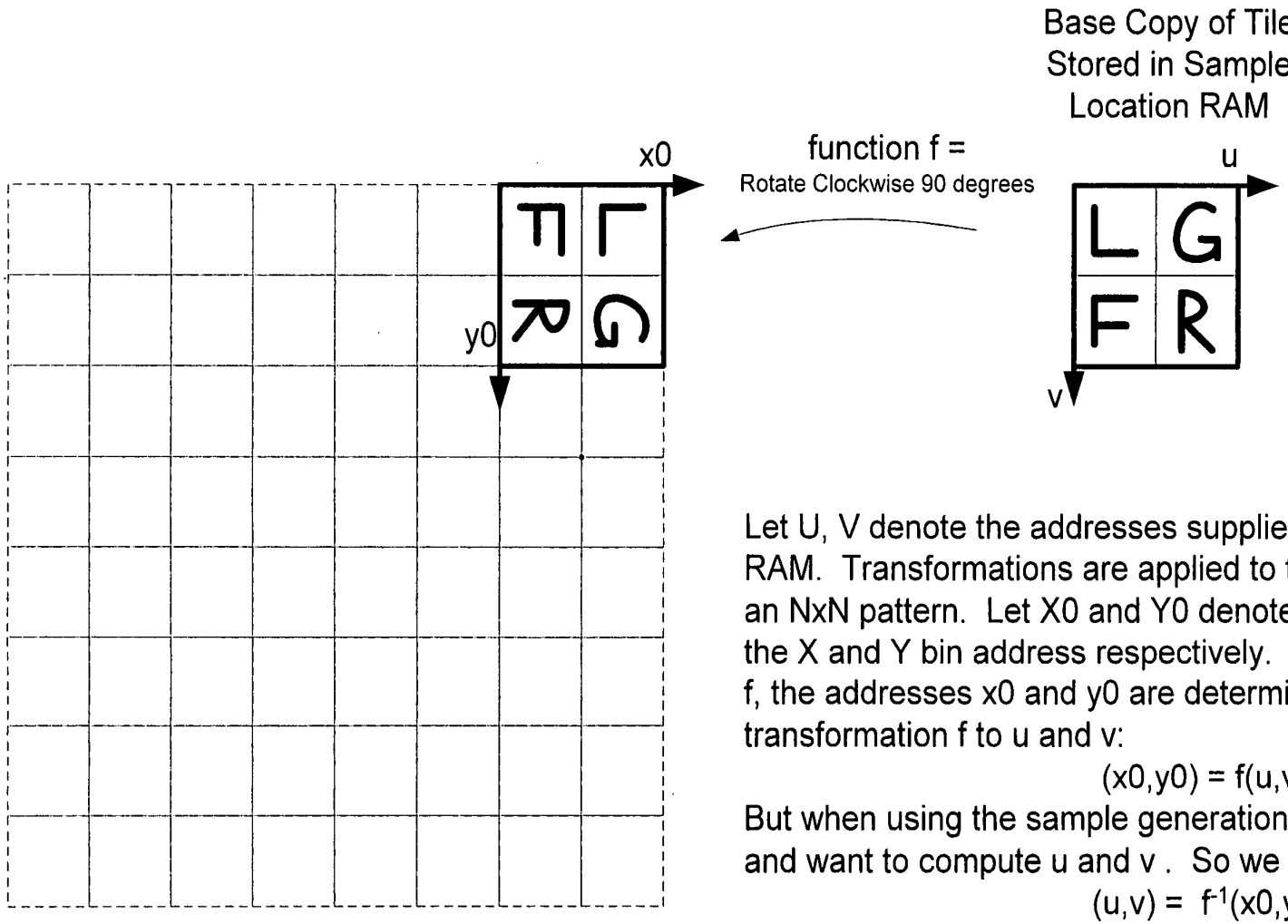

Bin Location
X,Y Kernels
Partition Location
X,Y Phase
Bin Sample Selector
1010
Permute Enable Signal
Temporal Permute Signal
Permutation Logic Unit
1000
Sample Location (X,Y)
Sample Location RAM
1020

Fig. 22



Let U, V denote the addresses supplied to the sample location RAM. Transformations are applied to the base copy to build up an NxN pattern. Let X0 and Y0 denote the least significant bit of the X and Y bin address respectively. For a given transformation f, the addresses x0 and y0 are determined by applying the transformation f to u and v:

$$(x0,y0) = f(u,v).$$

But when using the sample generation circuit, we have x0 and y0 and want to compute u and v . So we apply the inverse function:

$$(u,v) = f^{-1}(x0,y0)$$

Fig. 23

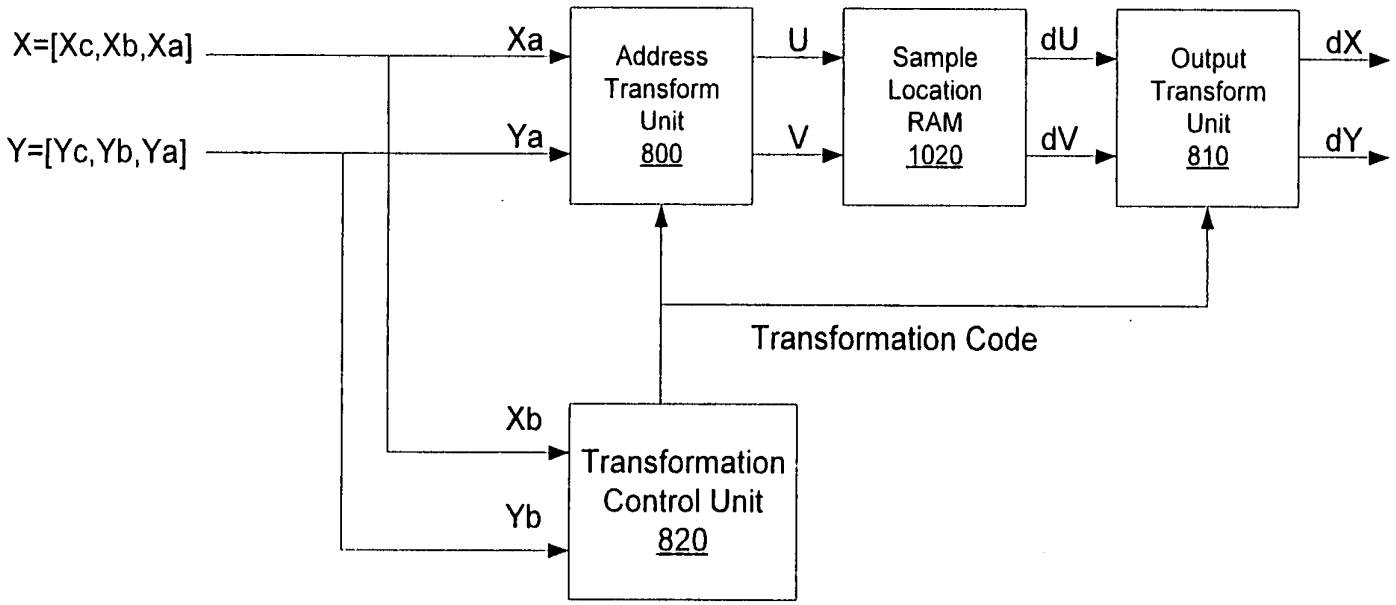

X=[Xc,Xb,Xa]
Y=[Yc,Yb,Ya]
Xa
Ya
Address Transform Unit 800
U
V
Sample Location RAM 1020
dU
dV
Output Transform Unit 810
dX
dY
Transformation Code
Xb
Yb
Transformation Control Unit 820

Fig. 24

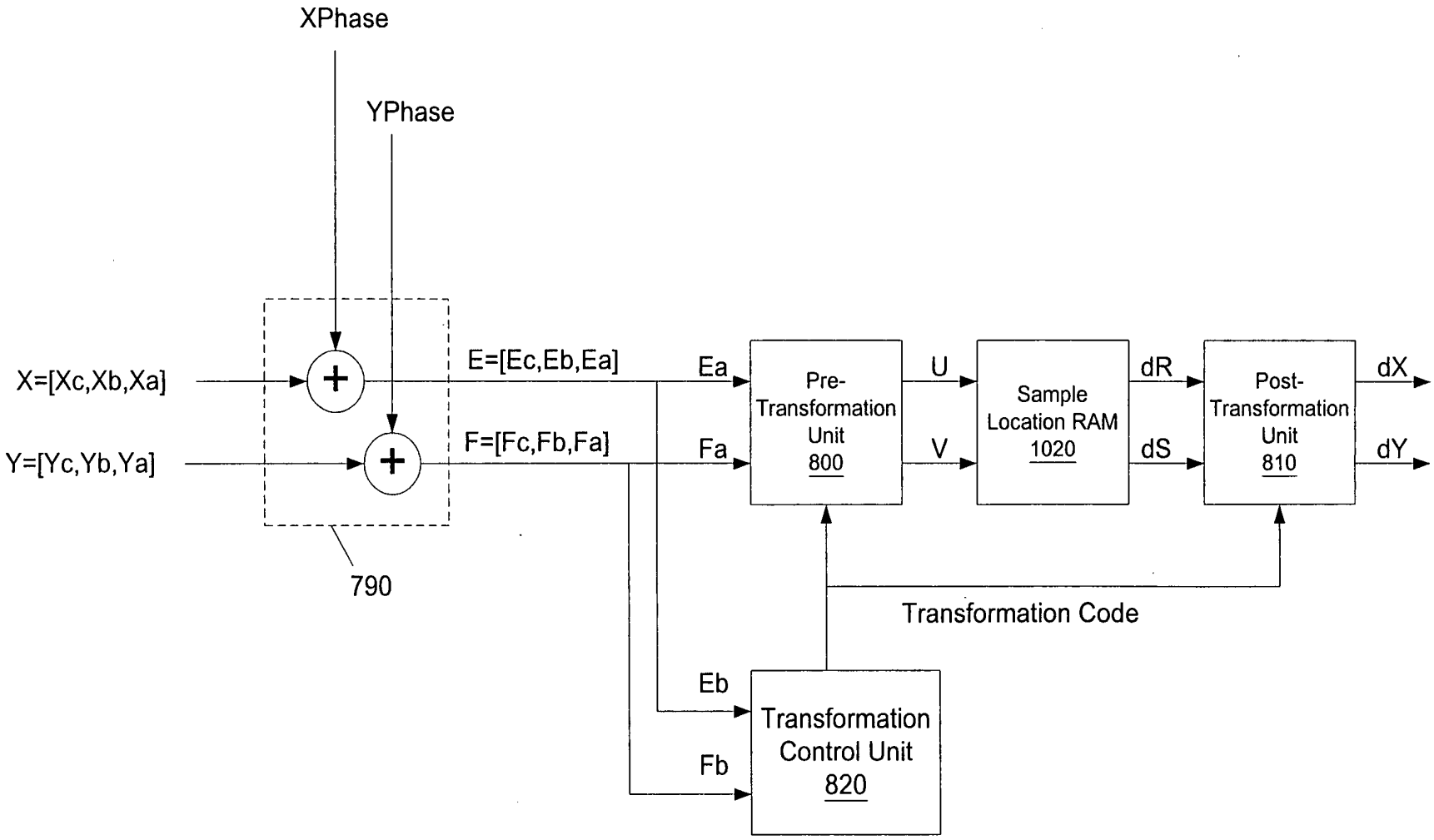

XPhase
YPhase
X=[Xc,Xb,Xa]
Y=[Yc,Yb,Ya]
E=[Ec,Eb,Ea]
F=[Fc,Fb,Fa]
790
Ea
Fa
Eb
Fb
Pre-Transformation Unit 800
U
V
Sample Location RAM 1020
dR
dS
Post-Transformation Unit 810
dX
dY
Transformation Code
Transformation Control Unit 820

Fig. 25

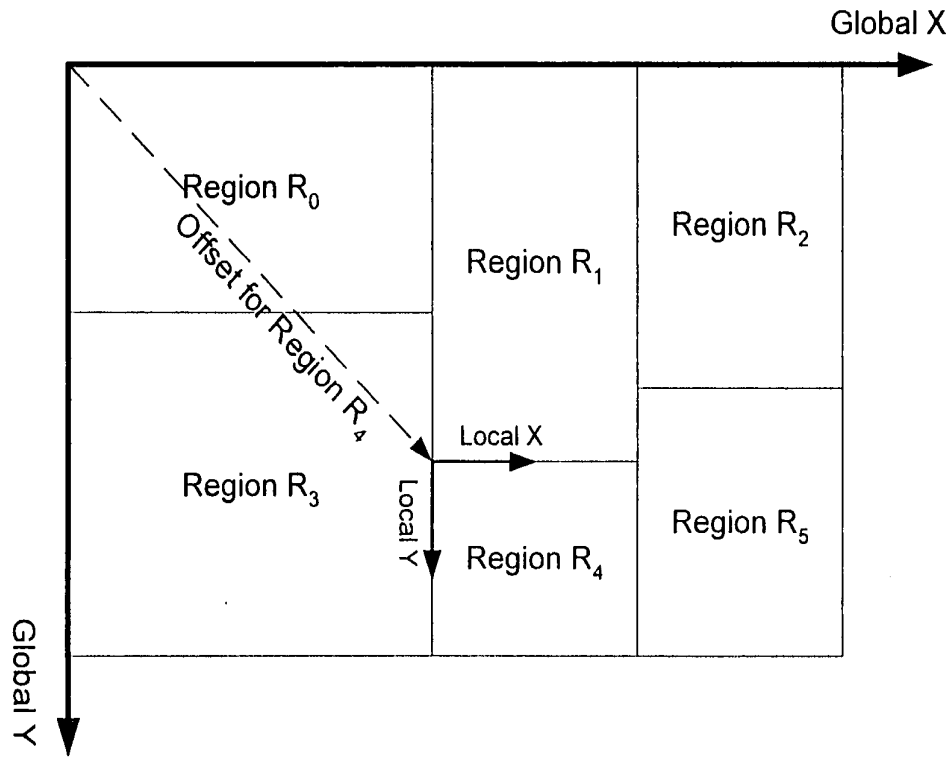

Global X
Region R_0
Offset for Region R_4
Region R_1
Region R_2
Local X
Region R_3
Local Y
Region R_4
Region R_5
Global Y

Fig. 26

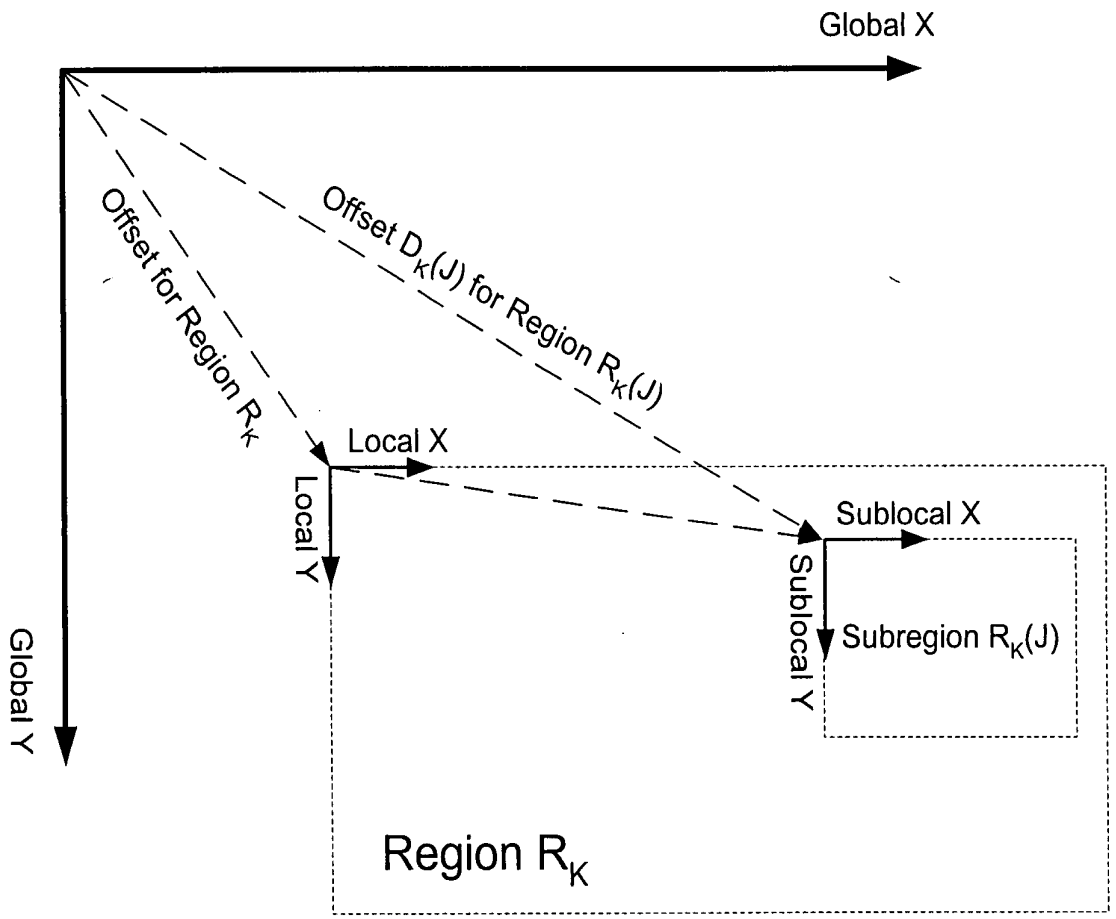

Global X
Offset $D_k(J)$ for Region $R_k(J)$
Offset for Region R_k
Local X
Local Y
Sublocal X
Sublocal Y
Subregion $R_K(J)$
Global Y
Region R_K

Fig. 27A

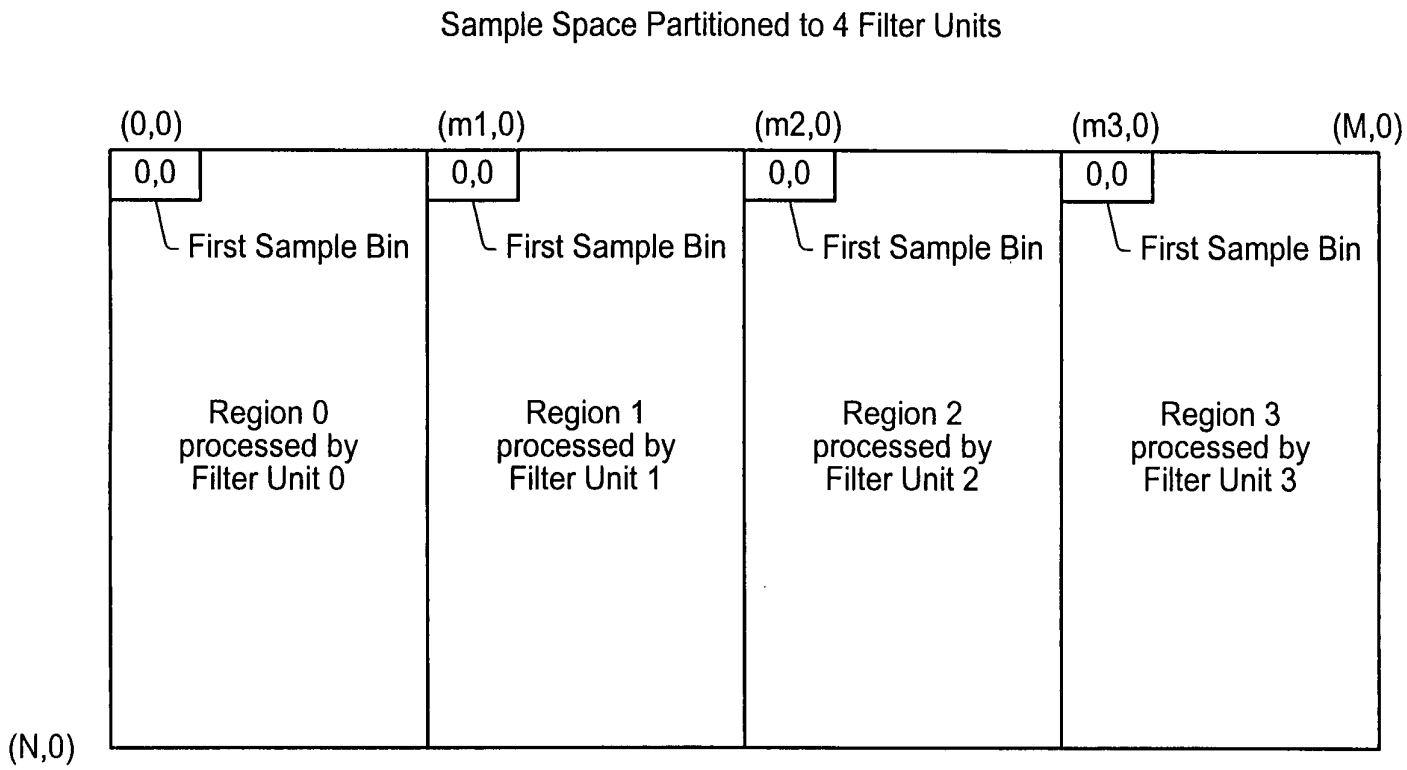

Sample Space Partitioned to 4 Filter Units
(0,0)
(m1,0)
(m2,0)
(m3,0)
(M,0)
0,0
0,0
0,0
0,0
First Sample Bin
First Sample Bin
First Sample Bin
First Sample Bin
Region 0 processed by Filter Unit 0
Region 1 processed by Filter Unit 1
Region 2 processed by Filter Unit 2
Region 3 processed by Filter Unit 3
(N,0)

Fig. 27B

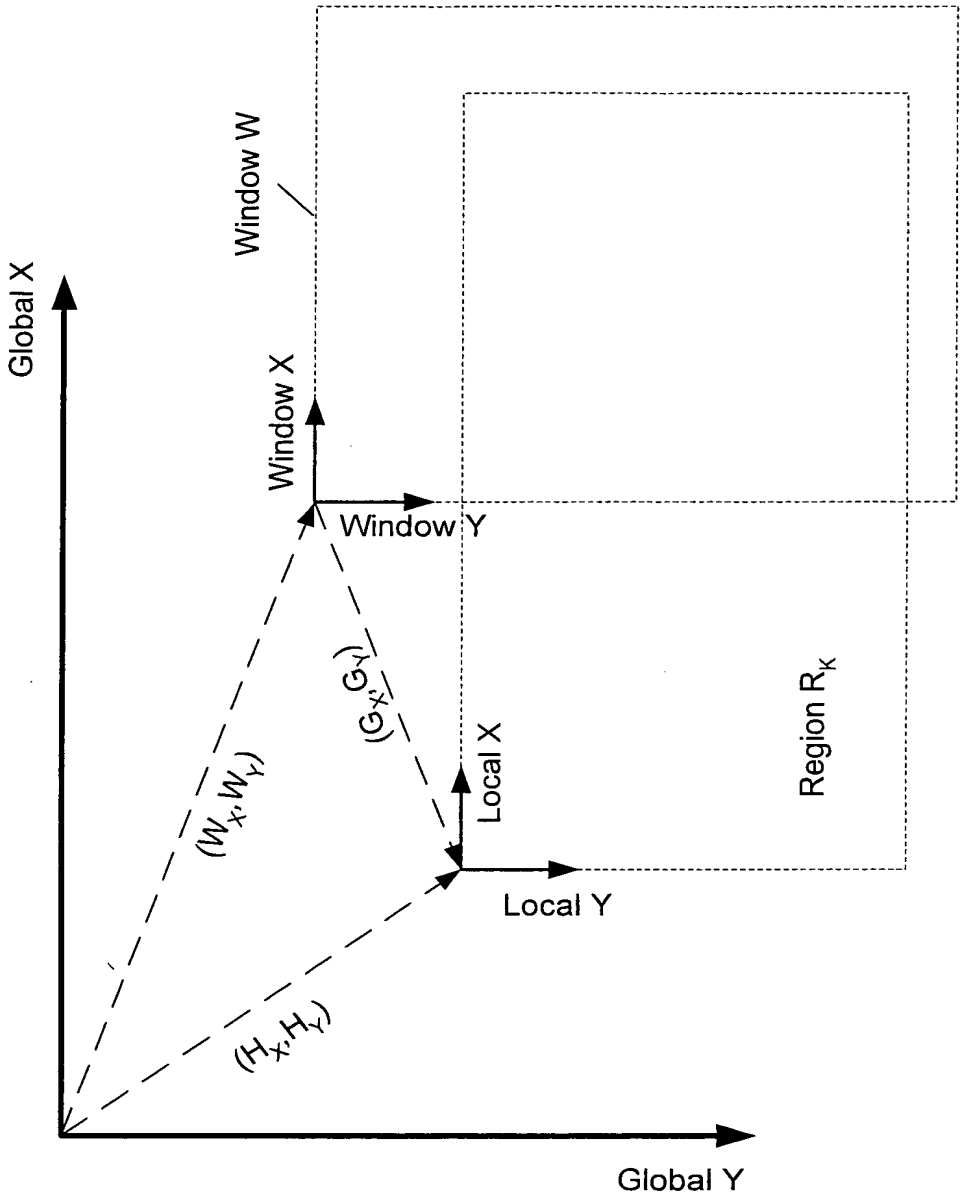

Window W
Global X
Window X
Window Y
(G_X, G_Y)
(W_X, W_Y)
Local X
Region R_K
Local Y
(H_X, H_Y)
Global Y

Fig. 28

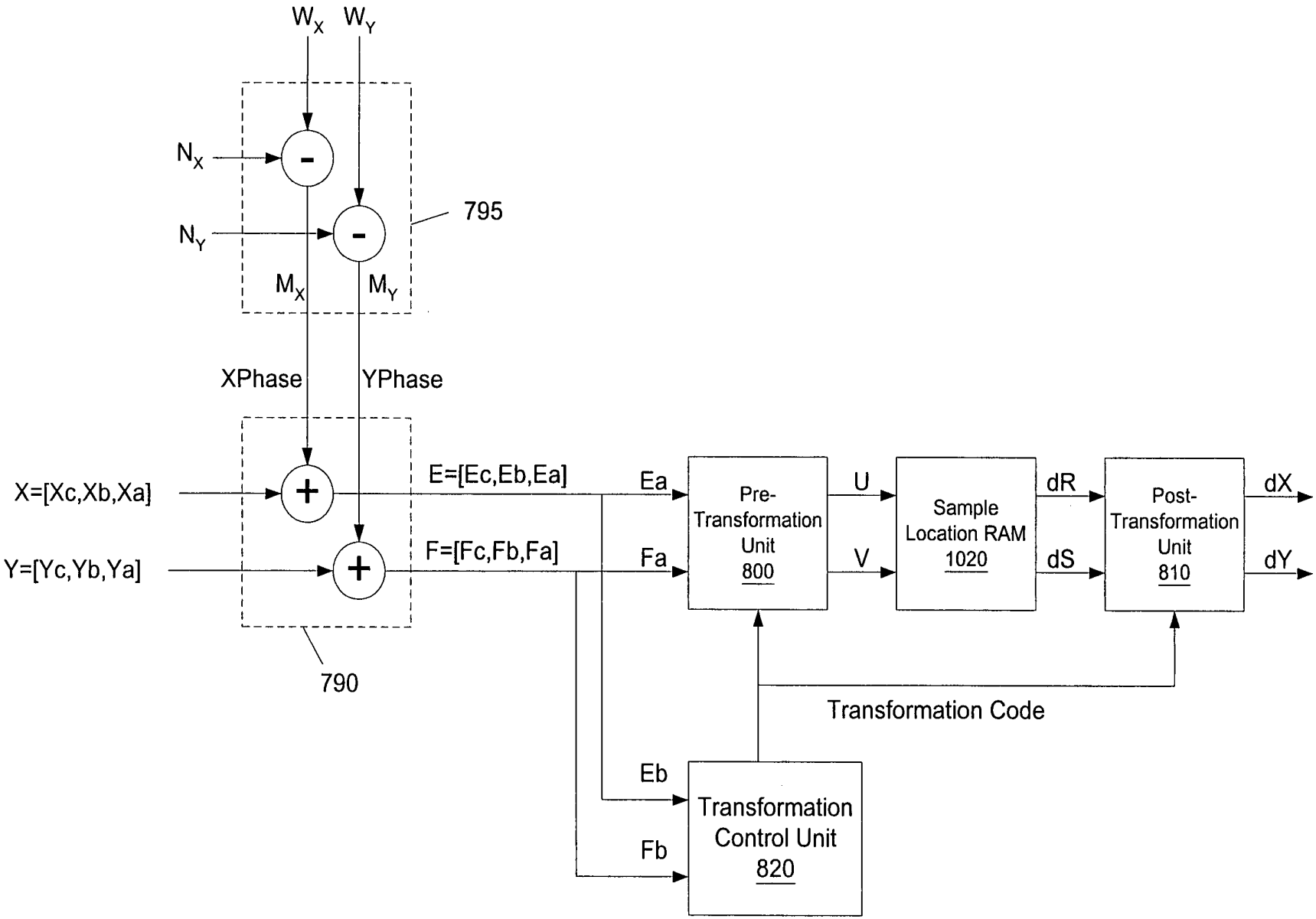

W_X
W_Y
N_X
N_Y
795
M_X
M_Y
XPhase
YPhase
X=[Xc,Xb,Xa]
Y=[Yc,Yb,Ya]
E=[Ec,Eb,Ea]
F=[Fc,Fb,Fa]
790
Ea
Fa
Pre-Transformation Unit 800
U
V
Sample Location RAM 1020
dR
dS
Post-Transformation Unit 810
dX
dY
Transformation Code
Eb
Fb
Transformation Control Unit 820

Fig. 29

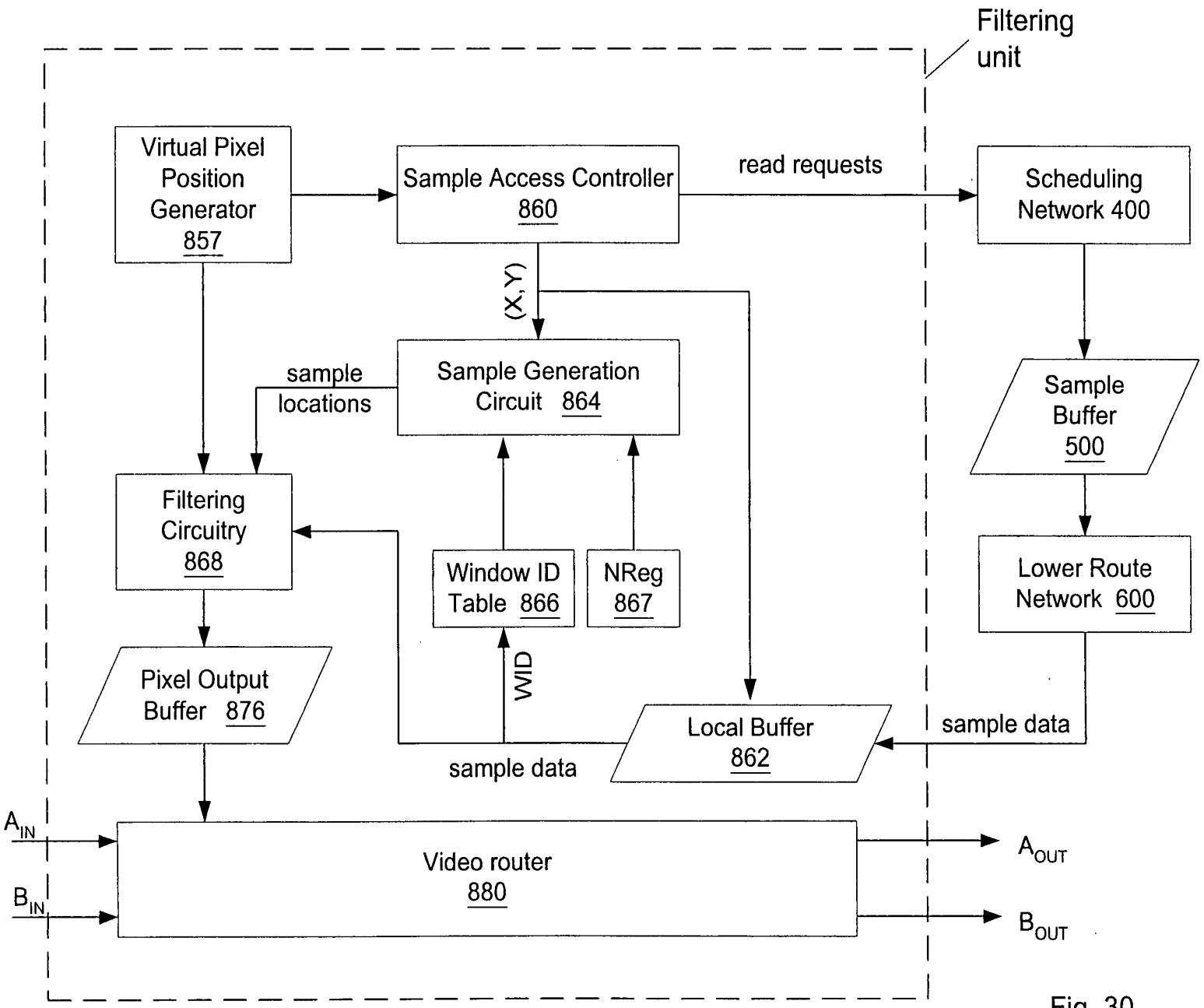

Filtering unit
Virtual Pixel Position Generator 857
Sample Access Controller 860
read requests
Scheduling Network 400
(X, Y)
sample locations
Sample Generation Circuit 864
Sample Buffer 500
Filtering Circuitry 868
Window ID Table 866
NReg 867
Lower Route Network 600
Pixel Output Buffer 876
WID
Local Buffer 862
sample data
sample data
A_{IN}
B_{IN}
Video router 880
A_{OUT}
B_{OUT}

Fig. 30